DO NOT REMOVE
FROM COAST LAB

CONTENTS

EDITORIAL STAFF

- Rob Kolstad, Editor
  <kolstad@usenix.org>

- Tina Darmohray, SAGE News Editor
  <tmd@usenix.org>

- Nick Stoughton, Standards Report Editor
  <nick@usenix.org>

- Carolyn S. Carr, Managing Editor
  <carolyn@usenix.org>

- Sylvia Stein Wright, Copy Editor

- Rhea Gossett, Proofreader

CONTRIBUTIONS SOLICITED

You are encouraged to contribute articles, book reviews, and announcements to *;login:*. Send them via email to *<login@usenix.org>* or through the postal system to the Association office. Send SAGE material to *<tmd@usenix.org>*. The Association reserves the right to edit submitted material. Any reproduction of this newsletter in its entirety or in part requires the permission of the Association and the author(s).

Working

Working is on my mind today.

Despite my strong embrace of the Puritan work ethic, I believe it is probably possible to work too much.

Studs Terkel's book, *Working*, features short vignettes of dozens of different kinds of laborers from throughout the workplace (e.g., salesman, steel worker, TV executive). Every one has the same message, as I recall: "No one understands how hard my job is. I work extra hard and do a better job than anyone knows." Many appended: "and I don't think people appreciate what a good job I do."

I must confess to harboring this same idea. So do most of the people with whom I work and a large number of people I meet on the telephone. I am no longer so sure it's an "engineer's thing" because both my parents (father: manufacturer's representative, mother: university job placement coordinator) have shared the same thoughts with me. Of course, I *know* I'm working hard, probably too hard.

How can you work too hard? You might put work ahead of play or other activities like seeing friends, taking time off, and physical recreation. Then bad things happen: a bad attitude, a tendency towards plumpness, and a feeling of lack of control is what happens to me.

I can always tell when I need a vacation because my impulse upon answering the phone is a cheery "This is Rob Kolstad; what the hell do you want?" I'm pretty sure Dale Carnegie would not approve of this scheme for greeting new and old customers.

So what am I going to do about it? As soon as I mail this off to USENIX for publication, I'm heading over to Colorado Springs' Lynmar Racket Club where I have an appointment to sign up for racketball courts and (gulp!) fitness machines. I'll be going there every night around 6:30. I figure that'll get me out of the office and enable me to return to my slim muscleman figure of years gone by (too many years gone by).

If you're working too hard and feeling unbalanced, I hope you'll find a way to force yourself to balance your life. I'll let you know if it works for me.

RK

USENIX members: in the coming
months watch *comp.org.usenix* for
an announcement about the Online
Member Survey.

OPINION

Update on Cryptography Issues

by Greg Rose
<ggr@sydney.sterling.com>

It's funny that when I put my foot in my mouth and started writing these articles for *;login:*, I really expected there to be a lot of movement over the next few months. Instead there has been a lot of interesting (to me) but dry (to everybody) action, almost all of which isn't at all easy to summarize. That said, here is my quick summary.

There is no apparent movement on the indictment proceedings against Philip Zimmermann. This is despite the beta release of PGP Phone, which uses PGP, a multimedia Macintosh, and a modem to give you a secure phone connection. In fact, I haven't heard much about people using PGP Phone, either. In any case, there had been speculation that this event would prompt the prosecution to move forward, but it seems not to have happened.

Within a couple of weeks of each other, two of the lawsuits that have been grinding away for some time moved another step. One is the suit brought by Phil Karn against the US Government for its refusal to grant an export license for the source code diskettes Bruce Schneier produced as an adjunct to his book *Applied Cryptography* (which is, by the way, now in its much improved second edition). Anyway, the government filed a motion to dismiss the case, Karn's lawyers filed a countermotion, and it will be heard soon. Philip Zimmermann even lodged a statement as part of the response. See *http://www.qualcomm.com/people/pkarn/export* for more on this one.

Roger Schlafly, attempting to get the patents on public key cryptography overturned, made a significant breakthrough. Mike Matyas at IBM Research had a date-stamped preprint of Diffie and Hellman's original paper in his files, dated more than a year before the patent was granted. His lawyers filed a motion for summary judgment in that case, talked about it in court, and this is being considered at the moment. This has nothing to do with export, really, but the patent stranglehold by RSA Data Security Inc. and Cylink Corporation (ex-partners in Public Key Partners) has had a chilling effect on development and deployment of commercial cryptography for some years.

On a lighter note, someone in England wrote a Perl program that can do arbitrary precision RSA cryptography in 4 lines of Perl. (Pretty cryptic Perl, mind you, but then what isn't?) He put this in his *.sig* file and pointed out that if you were in the United States, you couldn't quote his signature in email or news, because you would be exporting arbitrary precision RSA. Someone else picked up the idea of printing the four lines of code on a T-shirt, both in ASCII and in barcode. This makes the T-shirt both machine washable and machine readable. The latter of these makes the T-shirt itself a prohibited export according to any reasonable (!) interpretation of the ITAR (International Traffic in Arms Regulations). Variations of the T-shirt have been printed in the United States, England, and Australia. A request for export approval of the T-shirt has been filed, but the answer has not been received in the requisite 15 days.

[Editor's Note: News flash on January 12, 1996: The Justice Department dropped its case against Phil Zimmerman. No reason was given.]

Can UNIX Survive?

by Scott Hazen Mueller
<scott@zorch.sf-bay.org>

What a silly question. Of course UNIX is going to survive – in some form (Linux, for example) or another (Plan 9, for a different example). What we, as UNIX professionals, really care about is whether it's going to thrive and prosper. The alternative seems to be to learn Windows NT.

As I'm writing this, the much-ballyhooed launch of Windows 95 is just a few months in the past. Never before has an operating system upgrade received so much publicity. The multibillion dollar marketing machine known as Microsoft was very definitely in top gear.

The product itself seems to be technically mediocre. It attempted to lift some of the limitations self-imposed by Microsoft due to the design of DOS, a descendant of the much-lamented CP/M. The story I'm hearing is that the job is not complete.

On the other hand, the emerging story in the Microsoft arena is that Windows 95 is just a stepping-stone anyway. The overall strategy appears to be to get customers on to powerful enough systems that they can be migrated to the next release of Windows NT when it appears – late, if the company remains true to form.

The big UNIX-industry story is a little different than in the past, but in many ways more of the same. Novell bailed, not surprisingly, and sold the UNIX code base to SCO. The trademark and conformance suites had already been given to X/Open. At the same time, Hewlett-Packard – at Intel's urging – was put in charge of getting UNIX 64-bit ready.

With Sun already on an independent code base, it's hard to see how the industry is going to unify. Other vendors are being quoted as saying that they'll put on a happy public face, but carefully scrutinize the official UNIX work before taking it up. UNIX may have a lot of technical points in its favor, but it sure doesn't have anyone setting a forward-thinking and broad-based strategy for it.

And then, down at the bottom of it all, who do you find? Microsoft. Not only does it hold a stake in SCO, now the owner of the "official" AT&T-descended code, it also, according to reports several months back, gets a royalty for every copy of "real" UNIX sold, courtesy of a previous round of unification efforts. It looks to me like those folks can't lose, no matter which way it goes.

If you have the luxury of working in research or education, it probably doesn't matter that much to you if Microsoft comes to dominate the business world. After all, you can always port a copy of the latest UNIX or UNIX-alike to new hardware.

The rest of us, who work in business and industry, probably care a bit more. We'll have to live with whatever system dominates our workplaces. Some areas, like technical computing, will probably remain UNIX havens. I can easily see most other commercial gains being lost to Microsoft.

Of course, people have been crying the doom of UNIX for years. I don't think it's right around the corner, but I believe that it's coming. After all, UNIX's key advantages are only technical, and can be borrowed and copied.

Fixing the Wagon

What can we, the people who make UNIX tick, do? Well, we can do what we've been doing. It doesn't seem to have worked very well, though.

I don't claim to know all the key factors that would make UNIX successful in the commercial world. A lot of pundits have pointed to shrink-wrapped software as the big draw. Plenty of companies have chased that grail, with little success.

Other people point to cheap commodity hardware as the big key factor in a system's success. All I can say is you can get a decent UNIX free on dirt-cheap generic clone hardware, but it hasn't taken the computing world by storm.

Still others point to luck and timing. Although Gary Kildall wasn't really off flying the day IBM came calling at his door, looking for an OS for their brand-new personal computer system, Microsoft's place in the sun did have a lot to do with pure dumb luck. That doesn't mean Bill Gates hasn't made a fortune from exploiting that one opening, of course.

I do think that the free UNIX-like systems – they're not really UNIX if they haven't been X/Open certified – can be important. The folks who are working on them are doing a lot to dispel UNIX's image as difficult. They've got a long way to go, but they are making progress. I think that an easy-to-use PC UNIX, in combination with Wine (WINdows Emulator is one expansion), could make a decent dent in the desktop marketplace.

The high end is going to belong to HP. If companies do as they have always done and insist on "differentiating" their product in the marketplace with custom versions of UNIX, everyone is going to stumble. It behooves technical leaders to start rallying around a unified software platform. The premier CPU vendor in the world, Intel, has already publicly stated its concern that the PC software industry is just now starting to look to 32-bit software. They are ready to support

a competing operating system because otherwise their products are going to be hamstrung by the legacy of DOS and Windows 3.x.

USENIX members can take part in making this happen by bringing their companies to the table with HP and Intel, and really unifying UNIX. If it's not done this time, there may not be another chance.

Many thanks to the USENIX Corporate Sponsors!

Digital Equipment Corporation • Net Daemons Associates, Inc. • Hewlett Packard Co. • BBN Systems & Techs. • Intermetrics, Inc. • Timex • Sylvest Management Systems • Center for Communications Research–Princeton • Siemens Corporate Research • New Jersey Medical Underwriters, Inc. • Programmed Logic • Cexec, Inc. • Hazeltin Corporation • Lever Brothers Company • Fuji Capital Markets Corp. • The Sword Group, Inc. • Brookhaven National Laboratory • Stock & Associates • Northern Telecom • Alcoa Technical Center • Telerama Public Access Internet • Telebase Systems, Inc. • American Chemical Society Library • National Institutes of Health • SAIC/NCI-FCRDC: Science Application International Corp. • Infovision International • Circuit City Stores • Philip Morris, USA • MCDC • US Environmental Protection nAgency • Alacare Home Health Services, Inc. • Online Computer Library Center • Moen Inc. • State of Indiana, Information Services Division • Cimlinc, Inc. • Wisconsin Dept. of Transportation • UNISYS • Cray Research, Inc. • Rosemount Inc. • Platinum Technology, Inc. • Motorola Inc. • First Chicago National Bank & Dev. Corp. • IEX Corp. • Ericsson NetWork Systems • Rosetta, Inc. • Tandem Computers, Inc. • Bluegum Software Specialists • Idaho National Engineering Lab • 3M Health Information Systems • Intel Corporation • Mentat, Inc. • TRW •Amgen • Qualcom, Inc. • Nokia Mobile Phones, Inc. • AMCC • Indigo Services, Inc. • Sun Microsystems, Inc. • Genentech, Inc. • Advanced Micro Devices • Amdahl Corporation • McKesson Corporation • Digital Equipment Corporation Western Research Lab • Ipsilon Networks, Inc. • Fuji Xerox Palo Alto Laboratory • Pacific Bell • Open Vision Technologies • Western Micro Technology • Octel Communications Corp. • Hall Kinion, Inc. • Arca Systems, Inc. • Open System Solutions, Inc. • BT North America • O'Reilly & Associates, Inc. • US Department of Defense • Cascade Corp. • Willamette Industries • Walker, Richer & Quinn, Inc. • Boeing • John Fluke Mfg. Co., Inc. • Battelle Northwest • Message Handling System Pty. Ltd. • Radiophysics Laboratory Library, CSIRO • Mayne Nickless Courier Systems • The Land Information Centre, Department of Lands • Fulcrum Consulting • IIASA • Sobeco Ernst & Young • Sun Microsystems, CANADA • Department of Foreign Affairs & Intl. Trade Government of Canada • Data Kinetics • CSE • Hook Up Communications • SCO Canada, Inc. • IRS Computer Technologies, Inc. • Toronto Stock Exchange • City of Calgary, Data Processing Service Dept. • Dorchester Computing Ltd. • Nixu Oy • BULL • Institut Pasteur • Terminaux Postes Travail ICI IMA • Siemens Nixdorf Informations • PC-Plus Computing GmbH • SSO UNIX Software Support, Digital Equipment BV • Altos India Ltd. • Veritas Group • Siemens Information Systems Limited, Real Time Systems Division • HCL Hewlett-Packard Ltd. • Banca Popolare di Bergamo-Credito Varesino • Nippon Unisoft Corporation • Software Research Center, Ricoh Co. Ltd. • Surigiken Co., Ltd. • Access Co., Ltd • NEC Corporation • NTT Data Comm. Systems Corp. • Fuji Xerox • Hitachi, Ltd. • SISETSUKANG, Software Development Center, Hitachi • OKI Technosystems Lab. • National Computerization Agency • ETRI • Nederlandse Philips Bedrijven BV • Delft Hydraulics • Norsk Regnesentral • National Computer Board, Office & Network Technology Centre • Ericsson Telecom AB • Init AB • FMV: Intern ADB • Upsala Systemkonsvlt, UPSYS AB • DynaSoft, Dynamic Software AB • ENEA Data AB • CERN • Union Bank of Switzerland • IBM Zurich Research Laboratory • Goldman Sachs • ICL High Performance Systems • Swiss Bank Corporation, Info. Div. • Canon Research Centre Europe • Richard Shooter • AEA Technology (ISIS) • Technology Centre, Swindon UK • ANDATACO • Apunix Computer Services • Frame Technology Corporation • Matsushita Electric Industrial Co., Ltd • Shiva Corporation • Sun Microsystems, Inc., SunSoft Network Products • Tandem Computers, Inc. • UUNET Technologies, Inc.

President's Letter

10-4, Good Web

by Steve Johnson
<scj@usenix.org>

Those who study technology adoption like to draw curves of "percent penetration" of a new technology as a function of time. These curves have a familiar "S" shape – initially, only a few brave "early adopters" pick up the new technology. Then the curve becomes almost vertical as the technology grows like wildfire. Finally, only a few backwoods holdouts remain.

We are clearly at the vertical stage now with the Web. Almost daily, I am bemused by another company shoving its URL into my face; the Toyota home page appears at the end of an ad during a college football game. Our local racetrack has a home page. My realtor has a home page.

We are all well aware of the advantages of this technology, and some of these advantages are growing as "ordinary" people get on the net. My brother's "Happy Thanksgiving" letter was delivered in California a few minutes after he sent it from the East Coast, but the birthday card my in-laws sent my wife from Iowa took 9 days by snail mail. The middle school where my wife teaches is now "wired," adding a new flavor of messages to our home email: "Will the person who borrowed the ladder from the cafetorium please return it immediately, no questions asked."

However, there is often more glitz than substance in the net revolution. Grand schemes yield many missing links. Data from the early adopters has often not been kept up-to-date. Going to the Web to get the "latest information" is often going to disappoint.

I can remember only one other case in my life where a relatively abstruse technology, familiar to an in-group, was so suddenly embraced by the masses: CB radio. Younger readers may not remember this, but suddenly CB radio captured the imagination of the nation. Instead of being the working tool of truckers and taxi drivers, every Tom, Dick, and Harriet had to have a CB radio in the car. Articles appeared in the national press about how people could summon aid after a breakdown by using CBs, learn about the best restaurants in a town they were approaching, make new friends, even, in some well-publicized cases, meet the person they would marry. Comedians could get a laugh by inserting "10-4" into their routines. Does this all sound familiar?

Within a couple of years, CB was gone, almost without a trace. On the outskirts of many towns, you can still see a rusty sign proclaiming that the police monitor CB channel 9. For all I know, they still do. But a lot of the uses of the technology were replaced by cellular phones, which offered universal service, privacy, (believe it or not) better speech quality, and upwards compatibility with conventional phones.

So now we have the Web. It too is a kind of broadcast medium, and a lot of its appeal is the same as with CB – meet new people, get information from areas you are not familiar with, and even, for many companies, call the cops (customer

service) when your product doesn't work. At the same time, many of the things that killed CB are increasingly present on the Web as well: bad signal to noise ratio, loud and painful bursts of static, lack of privacy. Perhaps most seriously, the economic advantages of the Web are still very difficult to pin down (as they were with CB). This makes continued growth and development dependent on individual and corporate charity (or maybe you prefer the word "vision"), rather than economic advantage.

If the CB analogy holds, Toyota (whose home page has, among other things, a listing of fine arts festivals in the US) will see less and less reason to keep updating their Web page with information that doesn't directly concern their business. For most things email, which is more private and targeted, will replace the Web.

The exception would be if electronic commerce indeed became possible. Then people would have an incentive to keep the pages interesting and the information up-to-date. Unfortunately, the bars to electronic commerce are as much legal (who enforces contracts and collects the sales tax?) and social (which of several good security mechanisms should we adopt?) as technical. We live in interesting times.

1996 Elections for Board of Directors

by Ellie Young
<ellie@usenix.org>

The biennial elections of the Association will be held in March.

Ballots will be sent to all paid-up members on or about February 22. Members will have until March 28 to return their ballots, in the envelopes provided, to the Association office. The results of the election will be announced in *comp.org.usenix*, and in the June issue of *;login:*.

The Board is made up of eight directors, four of whom are "at large." The others are the President, Vice President, Secretary, and Treasurer. The balloting is preferential, with those candidates with the largest number of votes being elected. Ties in elections for Directors shall result in run-off elections, the results of which shall be determined by a majority of the votes cast.

Newly elected directors will take office at the conclusion of the first regularly scheduled meeting following the election, or on July 1st, whichever comes earlier.

As of this writing (January 8, 1996), no nominations from the membership (which were open until January 29, 1996) had been received. The following candidates for potential election to the USENIX Association Board of Directors were put forward by the USENIX Nominating Committee (see page 4 of the previous issue of this newsletter for their full report):

President	Andrew Hume, AT&T Bell Laboratories
Vice President	Dan Geer, Open Market, Inc.
Treasurer	Eric Allman, Pangæa Reference Systems
Secretary	Peter Collinson, Hillside Systems

Board directors at large (4 slots available):

Ed DeHart, CERT
Peter Honeyman, University of Michigan
Doug Kingston, Morgan Stanley
Pat Parseghian, AT&T Bell Laboratories
Margo Seltzer, Harvard University
Elizabeth Zwicky, SGI Corp.

We urge you to vote. If you would like to check the status of your membership, please contact *<office@usenix.org>*.

Need Tix for UniForum '96?

If you are interested in attending the UniForum '96 Conference and Exposition being held February 12-16 at the Moscone Center in San Francisco, contact the USENIX office for guest tickets: *<office@usenix.org>*.

Errata

Apologies to author Greg Rose whose article on PGP in the October '95 issue of *;login:* had its meaning changed in the editing process. The last paragraph of the article reads: "Finally, a trip to Australia hasn't made me immune to cryptography issues."

What Rose originally wrote was:

"And last, being in Australia hasn't made me immune from any of this after all."

Our apologies to author John Kohl for not including his name on the cover page of the December '95 issue of *;login:* as co-author of the article "The Net BSD Project."

SAGE
THE SYSTEM ADMINISTRATORS GUILD
NEWS

From the Editor

by Tina Darmohray
<tmd@iwi.iwi.com>

Everywhere I stop, look, or listen these days I hear things about NT. It's in the paper, splashed on magazine covers, discussed on the nightly news, and in my email box. I'd say it represents a pretty large bandwagon at this point. Despite the media attempts, I really hadn't paid much attention to the hoopla until a week ago when someone asked me, point blank, "How long until NT will replace UNIX?" I answered the immediate question by stating, honestly, that I really hadn't given it much thought. But I've been thinking about it ever since.

Suddenly, I wanted to know when NT was going to replace UNIX. I read ads, articles, tirades, etc. Nothing definitive. I polled friends, relatives, and my lover. I asked a roomful of people at a recent conference. The results were usually the same: some folks were convinced; others were dubious. Ultimately, I was left with a lot of opinions, but no clearer idea of when NT would replace UNIX.

Next I turned to history as a possible way to predict the future. I sat down to enumerate previous predictions of UNIX's demise. It wasn't long before I was chuckling to myself because I hadn't been able to list any times that "the end" was predicted for UNIX, but I could easily recall when the popular belief was that UNIX would never "make it." Gee, in the span of six years or so, UNIX had gone from certain failure to being replaced! (I guess that means it "made it" somewhere in between.) I finally realized that claims of overnight failure or success were more prevalent than actual instances of either. More often, success takes us by surprise. I thought of UNIX, X windows, the Internet, and Netscape stock.

I concluded, in all fairness, it's too soon in NT's life cycle to tell if it will change the entire computing world. But if NT does take off, its success will have been predicted, and UNIX's certainly wasn't.

SAGE Election Results

The results of the elections for four director seats on the SAGE Board of Directors for the 1996-1997 term are as follows:

Directors (Elected for 1996 & 1997, two-year term)

Paul Evans	321
Barb Dijker	314
Helen Harrison	227
Tim Gassaway	171

Not Elected:

Jeff Tyler	160
Kevin Kelleher	157
Peter Gray	145
Jim Campbell	117
Total Ballots Cast	456

Three directors (Bryan McDonald, Hal Miller, and Kim Trudel) will return to the SAGE Board of Directors for 1996. These Board members will complete their two-year term at the end of 1996.

The SAGE Board of Directors chooses its own officers after each general election every year (at its January 19 board meeting).

SAGE 1995 System Administrator Profile and Salary Survey

by Zanna Knight
<zanna@usenix.org>

For the second consecutive year, we conducted a System Administrator Profile and Salary survey at the LISA Conference in Monterey. 635 participants or 37% of the attendees responded.

Keep in mind that the data presented is based on LISA attendees, and not on all system administrators. While 635 is a respectable number of individuals, it is still a relatively small sample of the whole community of system administrators. Some of the survey results may therefore be due to the small size of the sample.

The survey respondents are 78% male and 20% female, representing a 2% increase in the number of females respondents over the previous year. 79% of the respondents work between 41-60 hours a week, and just 8% work from 61-80 hours a week, about the same as last year. 59% are compensated either by paid overtime or comp time, about the same as last year.

There has been an 8% increase of system administrators who always carry pagers, a 3% increase of folks who never carry them, and a corresponding 11% decrease in folks who sometimes carry pagers. The number of respondents who are compensated for carrying a pager doubled to 10%, and the percentage of respondents who are not compensated increased to 66% from 55%.

While the overall income distribution stayed about the same as 1994, there was a decrease of 4% in those earning less than $46,000 and a corresponding increase of 4% in those earning above $50,000.

With all data arrayed against distributed salary ranges, we were better able to view any gender-related salary differences. In 1995, 55% of the female respondents earned less than $50K versus 49% of the men, and 40% of the females earned over $50K compared to 51% of the men. This represents an improvement over 1994, when 66% of the women earned under $50K versus 50% of the men, and just 33% earned over $50K versus 47% of the men. Thus, there was an increase of 7% in females earning between $50-$75K annually and an 11% decrease in those earning less than $50K. Men in each of those categories remained about the same, within a percentage point. However, females have yet to break the $100,000+ category, and only 5% earned between $75-100K versus almost 8% of the males.

System administrators working in business/commercial environments are far more likely to earn $50K or more than those working for the government or educational institutions. In 1995, 32% of system administrators in businesses earn $50K or more (26% in 1994), compared with only 4% in education, and 7% in the government. There was a 25% increase in attendees working for companies with over 2000 employees. Those supporting more operating systems do earn more than those supporting fewer. There was a 6% jump in respondents who support six or more operating systems. 4% of the respondents checked a new job category, "independent/contractor," and of those 4%, three quarters earn more than $60K.

All SAGE members have been mailed the complete SAGE 1995 profile and survey. If you did not receive this mailing in December and/or are interested in receiving more detailed data, please contact *<office@usenix.org>* and we will send you the full charts.

Perl Practicum: Failed To Understand the Reference

by Hal Pomeranz
<hal@netmarket.com>

One of the nice new features of Perl5 is the ability to create references: a scalar that points to another Perl data object (e.g., a list or an associative array). Along with references comes the ability to create compound data types (lists of lists or arrays of lists, for example), which were difficult to create in Perl4. These new compound data objects have the typical properties of other Perl data structures – most importantly they automatically allocate storage for themselves, unlike C.

Some Concrete Examples

Perl5 adds a new \ operator to create a reference to an existing Perl variable. For example, here's how to create a reference to a simple scalar variable:

```
$scalar_ref = \$a_scalar;
```

When you want to get to the value of the scalar, you just substitute the reference for the name of the variable:

```
$$scalar_ref = "some value";
print "$$scalar_ref\n";
```

Note the double dollar signs. Perl uses the leftmost dollar sign to recognize what type of object we are talking about – in this case a scalar variable. With this information, Perl can appropriately dereference anything that might follow.

You can also create references to lists and associative arrays:

```
$list_ref = \@some_list;
$hash_ref = \%the_hash;
```

Again, the symbols surrounding the reference determine exactly how Perl will dereference and use the object. Here are a couple of examples using the list reference defined above:

```
@$list_ref = localtime();
$hour = $$list_ref[2];
```

In the first case we are resetting the entire contents of the list pointed to by $list_ref. In the second we are manipulating a single element. In the second case, Perl deduces the context from both the dollar sign to the left of and the square brackets following the reference.

The same idea applies to references to associative arrays, except the special characters there are % instead of @ and curly braces instead of square brackets:

```
%$hash_ref = (
    "January" => 1,
    "February" => 2,
);
$$hash_ref{"March"} = 3;
```

Things get even more complicated when we start having compound data types (arrays of list references, etc.). Suppose we were going to store various time vectors in an associative array. First we create lists holding the values, and then we store references for those lists in the array:

```
@gmtime = gmtime();
@localtime = localtime();
```

```
$time{"greenwich"} = \@gmtime;
$time{"localtime"} = \@localtime;
```

Sometime later, we want to get the hours value out of the lists. You might be tempted to do:

```
# WRONG! WRONG! WRONG!
$gmhour = $$time{"greenwich"}[2];
```

but this does not work. There is a precedence problem – scalar variables get dereferenced BEFORE key lookups. Because the scalar $time is undefined in our example, you will never get the value you want.

What you have to do is enclose compound references in curly braces:

```
# CORRECT
$gmhour = ${$time{"greenwich"}}[2];
```

The formal rule at work here is that you can replace a scalar reference with a Perl block – that is, an expression in curly braces. So the expression above is the moral equivalent of writing:

```
$list_ref = $time{"greenwich"};
$gmhour = $$list_ref[2];
```

This nested curly brace syntax is extremely cumbersome, so you can use the following shortcut:

```
$gmhour = $hash{"greenwich"}->[2];
```

C programmers should be familiar with the -> operator, which means "follow pointer"– same thing here. The left-hand side of the -> is an expression whose result is a reference, and the right-hand side is an index in the object that reference points to.

Because this is Perl, there is yet another way to do the same thing. You can omit the -> between list and array indexes (i.e., things in square or curly brackets):

```
$gmhour = $hash{"greenwich"}[2];
```

I generally prefer this last syntax, but your mileage may vary.

The -> was made optional for these operations simply because programmers commonly want to use multidimensional arrays and lists, and it is more natural to write

```
$coord[$x][$y] = $z;
```

than

```
$coord[$x]->[$y] = $z;
${$coord[$x]}[$y] = $z;
```

which are equivalent, but ugly and cumbersome.

Anonymous Data

You can actually create references to data objects that do not have an explicit identifier associated with them. This allows you to have static declarations for arrays and lists whose members are also lists (there was no way to do this in Perl4). Objects "created on the fly" in this way are generally referred to as "anonymous data objects."

The easiest cases are where we want to create an anonymous list or associative array and a reference to the object:

```
$short_months =
    ["Sep", "Apr", "Jun", "Nov",
        "Feb"];

$mail_info = {
    "hal" => "hal@netmarket.com",
    "tina" => "tmd@iwi.com",
    "rob" => "kolstad@bsdi.com",
};
```

So, square brackets for anonymous lists and curlies for anonymous hashes, just like their index brackets. These examples are not very interesting, however, because we could have just explicitly declared a list, @short_months, or an array, %mail_info.

Things get more interesting when we start declaring compound objects. Here is an example of declaring an associative array that has one value that is a list reference:

```
%hostinfo = (
    "name" => "myhost",
    "domain" => "netmarket.com",
    "addrs" => ["199.79.247.20",
        "204.25.36.200"],
    "owner" => "Hal Pomeranz",
);
```

You would print the second address with:

```
print "$hostinfo{'addrs'}[1]\n";
```

Yes, you can nest these kinds of declarations arbitrarily deeply:

```
@hosts = (
    ["name" => "myhost",
    "domain" => "netmarket.com",
    "addrs" => ["199.79.247.20",
        "204.25.36.200"],
    "owner" => "Hal Pomeranz",
    },
```

```
    {"name" => "thathost",
    "domain" => "netmarket.com",
    "addrs" => ["199.79.247.21"],
    "owner" => "Bob Smith",
    },
    # etc, etc, etc,
);
```

Given the declaration above,

```
print "$hosts[1]{'addrs'}[0]\n";
```

would print "199.79.247.21." Just to reiterate, you could also rewrite the above print statement either of the following ways:

```
print "$hosts[1]->{'addrs'}->[0]\n";
print "${${$hosts[1]}{'addrs'}}[0]\n";
```

You can see now why I prefer the first syntax.

Putting Things Together

Use of anonymous data structures allows us to simplify that "array of time vectors" example above. In that example, we explicitly created lists and then used the backslash operator to create scalar references to them:

```
@gmtime = gmtime();
@localtime = localtime();

$time{"greenwich"} = \@gmtime;
$time{"localtime"} = \@localtime;
```

Rather than creating the @gmtime and @localtime arrays, we could

```
@$gm_vec_ref = gmtime();
@$loc_vec_ref = localtime();

$time{"greenwich"} = $gm_vec_ref;
$time{"localtime"} = $loc_vec_ref;
```

This is not very exciting. True, we got rid of those annoying backslashes, but who really cares? Remember one of the early rules we learned: you can put a block in place of a scalar reference. This means that we can get rid of the extra assignment statements altogether:

```
@{$hash{"localtime"}} = localtime();
@{$hash{"greenwich"}} = gmtime();
```

We are just replacing $gm_vec_ref with the block {$hash{"greenwich"}}, and the same for the localtime() vector.

References to Subroutines

Because subroutines are just another Perl data object, you can create references to them as well:

```
sub hello {
    print "Hello world!\n";
}
$sub_ref = \&hello;
&$sub_ref();
```

Perl5 allows you to call your own subroutines without the &, but when you are dealing with references, Perl needs the & as a hint to tell it what type of data the reference points to.

You can also create references to anonymous subroutines:

```
$sub_ref = sub {
    print "Hello World!\n";
};
&$sub_ref();
```

Notice the trailing semicolon after the closing curly brace.

Other Useful Tidbits

Perl5 now has a `ref()` operator which tells you what kind of object a given reference points to. So,

```
$array_ref = \%this_hash;
print ref($hash_ref), "\n";
```

prints HASH. Other values returned by `ref()` include SCALAR, ARRAY (for lists), and CODE (for subroutine references). `ref($foo)` returns undef if `$foo` is not a reference.

By the way, the following code:

```
$refname = "foo";
$$refname = "Surprise!";
print "$foo\n";
```

prints Surprise! In other words, if you use a variable as a reference and if the value of that variable is not a reference, then Perl interprets the value of the variable as the name of an identifier. You can really shoot yourself in the foot with this one.

Coming in the Next *;login:*

Some of this reference stuff is mysterious at best, so in the next issue we will look at an extended example that covers all aspects of references discussed thus far. Here is the problem, so you can have something to practice on in the next couple of months.

In my last column I briefly mentioned the concept of "marshalling" data: converting complex data objects to a format that can easily be saved to disk and retrieved later. The idea is to create a function `marshall()` such that if we

```
$string = marshall($some_ref);
eval("\$other_ref = $string");
```

then the data structure pointed to by `$other_ref` will have the same contents as the data structure pointed to by `$some_ref`. Remember that the data structure pointed to by `$some_ref` could be arbitrarily complex: a list of associative arrays whose elements could be lists, arrays, and/or scalars, for example.

Good luck with your coding. See you next time.

Tcl, the Other Applet Language

by John E. Schimmel
<jes@sgi.com>

For some months now, the hype over Web applets has been sweeping the industry, and the term "applet" has so far been synonymous with the Java language. Java itself is a reasonable language, but so far it does very little. Having Java be useful requires people to port all their multimedia libraries to this language or to provide stubs. There is another safe interpreted language already in existence, supported on dozens of operating systems including Windows and Macintosh, with several years worth of graphics content. This language is, of course, Tcl.

The strength of Java as an applet language stems directly from the fact that it is a very limited language. There is no support for pointers, and in theory, no way to directly trash the client system. Of course, if the content libraries contain a bug that would trash a system, it would certainly allow that to be exploited.

Tcl does have built-in support for getting new memory, reading files, etc., but Safe-Tcl, the result of some work done originally by Nathaniel Borenstein, allows configurable limitation of the Tcl language by pairing two interpreters and purposely deleting any dangerous commands.

A second strong point for Java is that it is faster and compiles into an intermediary language that is faster to download and is smaller and easier to interpret. In practice, however, using Hot Java and Netscape as examples, I find that the Java interpreter implementations are far from fast and efficient.

The strengths of using Tcl as the Internet scripting language is that it already runs on practically every system in existence. Tcl includes Tk, a set of portable graphics widgets. And Tcl has been around long enough to have dozens of existing extension libraries.

The choice of Netscape to support Java has tremendously accelerated the rate of porting the language to new hardware platforms and operating systems, but for the most part, Tcl already runs there.

The Java language has no support for bringing up windows on the screen, for playing movies, for doing anything that you would really want an applet language for, in a portable way. In fact, currently it cannot even create a new HTML page for the browser to display because that is seen as a security problem. Many individual companies such as Sun and SGI have added Java stubs to their existing media libraries, but it may take years of arguing in standards meetings for an extension API to exist. Again, for the most part, these already exist with Tk.

At the last Tcl Workshop (Toronto), Steve Uhler from Sun demonstrated a Web browser written entirely in Tcl, and supporting Tcl applets. At the time, he was calling the browser Hippo, short for HippocriTcl. Since that time, the browser has been fine-tuned and released as SurfIt! by Steve Ball.

With all the current hype around Java, I have no doubts that the needed support libraries will be added to the language to support all of the multimedia content in existence, but in the meantime I strongly suggest picking up a copy of the SurfIt! browser and playing around. You may be surprised at what you can accomplish while waiting for your Java development environment to arrive.

For the latest information on the SurfIt! Web browser, visit the Web page at: *http://pastime.anu.edu.au/SurfIt/*. For current information about Tcl, visit: *http://www.sco.com/Technology/ tcl/Tcl.html*. And, as always, watch for future editions of this article for what is hot in Tcl.

Elizabeth's Advice for People Who Answer Telephones

by Elizabeth D. Zwicky
<zwicky@usenix.org>

These are my 10 commandments for answering telephones. I'd love to say that I always follow them, but the best I can say is that I always try.

1. Never lie. In particular, avoid the temptation to say "You can't do that" if what you mean is "I don't know." It's also very easy to say something you're not sure of and then get sucked into defending it with complex rationalizations. Most people, even when they don't understand what you're talking about, will get an uncomfortable feeling that you're trying to snow them. The ones who do understand what you're talking

about will instantaneously write you off as a lying jerk who's not even smart enough to know what you don't know.

2. Trust criticism. People who say critical things are usually unfair and often whiners who complain all the time, but they are also usually partially right. You should feel absolutely justified in saying that they are mean, nasty, don't comprehend the problems of your job, wouldn't acknowledge outstanding service if they got it, and don't deserve it in the first place. When you're finished with that, go back and pay attention to what they're saying because 99 times out of a hundred you genuinely screwed up, and you need to fix something.

It's like not wearing your seatbelt; most of the time you can forget your seatbelt and nothing will happen to you. Sometimes you don't wear your seatbelt and you die. Nobody deserves to die because he or she forgot to wear a seatbelt. It's completely unfair and out of proportion. Nevertheless, not wearing your seatbelt is stupid. Most of the time you can make customer service mistakes – forget a call for a week, hurry a customer to get him off the phone, make up stuff you don't know, work in an area you don't understand, talk down to customers, rearrange things to make your life easier even if it makes theirs harder – and they will be so glad to have gotten any help whatsoever that they won't complain. Some of the time, they say you are not only a bad employee, you're probably a bad person, and you ought at least to be fired and probably shot. This is completely unfair and out of proportion. Nevertheless, making these mistakes is stupid, and ignoring the complaints is downright moronic.

3. Never turn off your brain. There are two situations where it's easy to turn off your brain and go on autopilot. First, when you have a common problem, it's easy to listen to key words in a problem description, decide it's the problem you're used to, and apply the solution for it without actually thinking deeply about what the customer is asking for. Customers describe problems in unexpected ways, and they may inadvertently use a description that kicks you into autopilot when they want something completely different. Often you end up trying to fix something that's not even broken, which will frustrate both you and the customer.

It's also easy to turn off your brain because you hear an unfamiliar key word; somebody calls about a program you've never heard of or is running an operating system version you're not used to, and you make the blanket assumption that you don't know and you can't help. In fact, experience on one program or one operating system generalizes pretty well, and you probably have more experience and more flexibility than the caller. Looking at the problem briefly may let you solve it. (The downside here is that the user will think you are the world's expert on the application because you were able to guess that "Save" was probably on the "File" menu.)

4. Put yourself in the caller's place. Yes, it's annoying to deal with someone who's so panicked and uncomprehending that they can't cope with questions like "Is there anything showing on the screen?" But there are probably situations in your life that you find equally intimidating. Try these:

Your car has started to make this noise, intermittently, after you've been driving it for about 30 minutes. It's something like a combination of a donkey braying and glass marbles being dropped out a second-story window. You can't describe it, you can't imitate it, you can't cause the car to make the noise at will, you're not certain where it's coming from, but you're pretty sure that when the noise happens, you also are getting a faint smell of something burning so you have taken the car to a mechanic, who is giving you a completely unbelieving look and seems to be trying to tell you that it can't possibly be doing that, but if it were doing that, it would be liable to catch on fire at any moment.

You are at an extremely fancy dinner. You have 17 eating implements, or at least you think you do, but there are three that don't seem to be knives, forks, or spoons, and you're not entirely clear what they might be. You have just been served something that closely resembles Jello, and the rest of the table appears to be waiting, with growing impatience, for you to start eating it. Or maybe it's not edible at all, and they're waiting for the food to be served. But an awful lot of them seem to be looking at you. . . .

You have an important presentation to give in 15 minutes, and you are copying it onto overheads. On page four of 120, the copier has jammed. You're not certain whether the smell of melting plastic is growing stronger, but it's certainly annoying. You're staring at the diagram inside the copier, which seems to want you to do something to a lever marked "4." Unfortunately, there's no context pictured around the lever, and you can't find anything marked "4." You have manipulated everything you can find that does have a number on it, although several of them didn't seem to move, and two of them moved quite a lot and don't look like they went back to exactly where they started. Also there was a snapping noise when you turned one of the knobs. There is no overhead visible, but there does seem to be a lot of toner about, your hands are getting pretty black, and you are starting to have visions of appearing in front of a company CEO with a large smudge on your nose, a permanent black stripe on your shirt, and no overheads – not to mention having to explain how you destroyed the copier.

If you can imagine how you would feel in any of those situations, and you are not disgustingly self-confident, you can imagine the peculiar emotional state many people are in when they call. They want you simultaneously to do something useful and to reassure them that they are not horribly stupid people with problems that no one of average intelligence could ever have. They are probably perfectly nice, normal human beings when the world is not conspiring to torture them, but

right this moment things are going to hell in a handbasket. Plus they are torn between thinking they are being completely stupid and there is some perfectly simple solution that is going to be humiliating to have pointed out and thinking they are facing irretrievable disaster. They don't like either option.

5. Never trust customers on matters of fact. Not even "My system is down."

6. Always trust customers on matters of opinion. They're entitled to their own opinions and if they want to do it upside down and backwards because it works better numerologically, even though it takes them four times as long, it's their life. They're entitled to hate perfectly nice software programs, to think that the latest and greatest version is putrid, to want their screen purple and green, or to refuse ever to change anything from the system default. Don't argue about it; at all costs don't tell them that they really *do* like it; and never, ever, ever, decide that they were just kidding and they'll thank you for installing the newer, nicer version.

7. Do not criticize or contradict other helpers in front of a customer. "Gee, what idiot worked on this last?" is a bad move. If the idiot in question witnesses the exchange, you have forever damaged your working relationship with this person, and even if she was being completely idiotic, you are not going to be able to discuss it rationally. Even if the person being criticized isn't there, you're encouraging the customer to be nasty. Doing this may make a customer bond to you, as a soul sympathetic to his plight, or it may just make him unhappy about the whole experience. What it won't do is help get his problem fixed. If the previous person who worked on it was an idiot, you need to get that dealt with – you may even suggest to the customer that he discuss it with the idiot or the idiot's manager to see what's going on – but you don't need to sabotage someone who, like you, is trying to do a difficult job, is probably overworked, and has better days and worse days. If you really want to make everybody's life difficult, there's always "Oh, we have a lot of trouble with that group. I don't think they really do much, and they always do such a poor job when they get around to working on something." This will get back to the group in question either as "Everybody knows you're no good" or as "The help desk people hate you." Either one will make them feel miserable and oppressed without giving them any information on how to improve.

Similarly, don't argue in front of a customer. Customers want to feel like they're dealing with people who know what they're doing. "No, I think it's the red wire that will make it explode" is not the kind of thing they'd like to hear. If someone is trying to give an answer that's functional, but wrong in the details or suboptimal, it may be best to let it go and correct it after the call is over. Obviously, if a colleague is saying, "Oh, `rm -rf /` is a *great* way to get a little free disk space," you may need to intervene extremely hastily, but do it with as much grace as possible. Scream "What kind of a moron are you, anyway??!?" only when no customers will hear you.

8. Stay calm. Keep breathing, slowly and effectively. Take breaks, move around, do what you need to stay focused on the world at large and not just one customer's little piece of it. When it gets too much, move furniture, throw things, jog around the building, go to the bathroom and cry, laugh hysterically, whatever works for you. As long as you keep your grip, nobody here is going to die. Besides, the customer is probably tense enough for both of you.

9. Know what you're doing. Don't memorize procedures without knowing at least roughly what they do and how, and use voodoo system administration only as a last resort. Logging out and rebooting may work much of the time, but they have significant side effects, they are often only temporary fixes, and sometimes they make life much, much worse. In general, applying random procedures is a terrible time waster and often leaves the user wondering about your competence. If you don't know how something works, you can't actually fix it. (However, some days the machine *is* inhabited by evil spirits, and voodoo may be appropriate. The important thing is to be able to tell this from a machine that's acting up for mundane reasons.)

10. Be flexible. There are no hard and fast rules. Do the right thing for your employer, your customer, and yourself. If that requires breaking rules, making up policy, and standing on your head, so be it.

Stress Relief

by Steve Simmons
<scs@lokkur.dexter.mi.us>

Who hasn't had the urge to put a brick through the front of the CRT? OK, you can all put your hands down now – and you two who didn't raise them are fired for lying.

It would be *so* satisfying to put a brick through your screen. But it tends to be kind of a one-time thing (well, one time per employer) and is rather dangerous – if you've never seen a big CRT go, consider yourself blessed. Still, who hasn't at least fantasized about it? I certainly did. But rather than destroy equipment and my career, I would go blow off steam in the kindly ear of Kathy Schneider. Kathy, among other things, was a good person to calm me down and put my head back on straight. She introduced me to the CRT brick.

CRT bricks, as supplied by Kathy, were chucks of yellow foam rubber roughly the size and shape of a brick. The sides were proudly annotated "CRT Brick," and it was a darned good thing because nobody would ever mistake this thing for a brick. But it was handy for throwing at screens, printers, PDP-11s, your office mate, and anyone who needed a brief, nondamaging interruption. Kathy would award you a brick if you complained with enough intensity, validity, and creativity. And she had fairly high standards.

When I left that job, the brick went with me. Eventually, I gave it to someone who needed it more than I, and for a long time that was the end of it. One day I noticed that large Sun monitors were shipped with protective foam in long rectangular chunks. A quick slice and application of magic marker and voilà, CRT bricks. I gave them out to my staff, and they began to spread. We gave them out to people who reported interesting problems. We gave them out to people who complained about Sun. We gave one to the vice president's secretary, who kept it in a drawer where he couldn't see it. We started having people beg for them. It turned the day-to-day complaints into entertainment and made everybody's job easier.

If we'd gone down to the local novelty store and bought rubber bricks, this probably wouldn't have worked nearly as well. Part of the point of the CRT brick is that it's so obviously silly. One person was quite insulted when I gave him the brick and thought I wasn't taking him seriously, so I offered to let him throw it at me. Later he became more understanding and even used it – judiciously, of course.

What Do Politics Have To Do with System Administration?

by Wendy Nather
<wendy@il.us.swissbank.com>

Answer: rather a lot, really.

All of you who claim you're not into politics because you're a technical person, please raise your hand.

Politics have nothing to do with the technology itself; that's true.

Computers are not political. However, there are still humans operating the computers. So you can't avoid politics.

Politics tend to center around four basic issues: who pays for the resources, who gets the resources, who controls the resources and their use, and who gets blamed for the problems.Here are some of my favorite political issues that crop up in my life as a system administrator:

1. Who paid for this machine, anyway?

Humans are territorial creatures, and machines cost money. Any time a machine is reallocated, moved, decommissioned, upgraded, or otherwise has its finite resources shifted to someone else's benefit, you as the administrator may land in the middle of an argument between users, department heads, and accountants – which is why asset management and tracking are so very important. I have the

benefit of an internally written asset management system that enables me to settle disputes with a simple report: the history of the equipment and who's currently paying the bill for it.

2. Who controls changes to the systems?

This issue provides constant fuel for flames, especially if you have support responsibilities divvied up among several groups, each under different management. Who gets to change the system, and who gets blamed if something breaks? I have a favorite answer to this one: the person who is to be held accountable is the person who has the control. Period. Anything else is not fair to the system administrator. You need very clear and definite backing from your top management to implement it, however, and interpretation can be a full-time job. This issue leads quite nicely into the next one.

3. They don't know what they're doing!

Perhaps the most common source of politics in any field is that. Everyone believes she knows what she's doing and others disagree with this self-evaluation. But technical incompetence is very hard to prove and requires a lot of experimentation on a system that may not be able to take the strain. So people have to fall back on innuendo rather than walking into a manager's office and saying outright, "This person is a bozo and should be fired immediately." It usually takes several giant technical blunders to prompt managers to solve a competence problem.

4. I'm never your highest priority.

This complaint can come regularly from your users if you are at all overworked (and who isn't?). Someone has to be at the top of your list, and someone has to be at the bottom. Deciding who is where on the list may be your decision, or you may have the decisions made (and regularly changed) by your management. Even if you have rules on priorities (for example, production users come before development users), there are people who won't be happy with their spot in the food chain, and I have seen dissatisfied groups try to break away and hire their own dedicated system administrators to solve the problem. The only solution I have seen that even comes close to working is to explain to management that if a group is allowed to hire its own resources, that's money that will be spent by the company anyway, and they might as well allocate it to you instead to improve your own resources. If you can improve your response time sufficiently, users won't care as much whether they're first in line or last in line, as long as they're getting served within a reasonable and consistent time frame.

Another solution I have heard about is the practice of regularly reviewing all items on your "to do" list and upgrading the priority of those that have been waiting a long time (no

matter what priority they started at). If enough time goes by with this method and ALL your items are at the highest priority, then you know that you *definitely* need more resources.

5. What are the politics of security and privacy?

Enough has been written about this topic for several books, so I won't go into it here, except to say that the Golden Rule still applies: whoever paid for the machines (see number 1) gets to make the security and privacy policies.

6. Who's supposed to clean up this mess?

Deciding who is responsible for what is another political nightmare. If you're unlucky enough to be in a corporate culture where everyone insists on his God-given right to go home at 5:00 pm, you may get stuck with a lot of the dirty work. Taking over too much of what no one else wants to do will land you back into the control issue as soon as someone notices you're doing his job.

Managers may try to take all these issues out of the hands of your employees in the hope of saving them the time and annoyance. But politics are inherent in any job, and you're much better off encouraging each system administrator to be politically astute rather than trying to shield them. Technical decisions are made every day based on politics, for the reasons above; and if system administrators are to make decisions, they need to understand *all* the factors rather than just the ones and zeroes.

System Administration Employment from a Student's Point of View
by Jeff Allen
<jeff@hmc.edu>

As a recently graduated student, I have a special interest in the employment picture. The demand for well-trained system administrators is soaring as the Internet becomes a household phrase. The cost of a protracted talent hunt, or worse yet, the cost of getting stuck with a poor system administrator, makes it worthwhile to make an investment in the rising generation of system administrators.

Much discussion has taken place in the system administration community about training, certification, and professionalization. The fact is, most system administration training still happens on the job. This phenomenon of apprenticeship is not really the problem – as a matter of fact, I have found it to be a very satisfying way to learn. The problem is that compared to the sky-rocketing demand for

system administrators, there are not enough of these apprenticeships available. While attempts to create effective system administrator training programs and certification programs progress, there are four things you can do in your company, *today*, to start making a difference in the success of your hiring efforts:

- Support USENIX's efforts to recruit students.

    LISA and the USENIX Technical Conference are excellent places to meet students and make contacts. However, many students can't afford the travel, hotel, and registration fees. USENIX has long made a point of including students by granting travel/housing stipends. A few years ago, I was lucky enough to receive some of that money, which enabled me to go to LISA VII. As a result, I made contacts with several companies, one of which resulted in an internship with Synopsys.

    Last year, USENIX awarded 46 travel grants to students. Recently, the student stipend fund has been increased from $30,000 to $150,000 per year (*;login:*, December 1995, pp. 24-5). If you agree with me that this is a great way for USENIX to serve its members, join me in thanking the USENIX board. If your company would like to help contribute to this student fund, I'm sure USENIX would be happy to accept a gift.

- Get connected to your company's recruiting efforts.

    Every year, over 50 moderate- to large-sized companies visit Harvey Mudd College looking for new employees. They hire for all kinds of jobs, and from all kinds of majors. The one constant, though, is their *disinterest* in system administration experience and skills. I have many times talked to on-campus recruiters who have no clue how they would go about turning over a qualified system administration candidate to their company's system administration group.

    My message here is simple: the infrastructure for college recruiting is already in place in big companies; *use it* to get fresh meat for your in-house training programs.

- Set up internship programs.

    Here's one suggestion for such a program, which comes partly from personal experience: find a project that would help your group out, but is non-critical to the operation of the business. Have an intern spend about 20 hours a week on the project. Use the rest of the intern's 20 hours a week as your personal assistant. Both parts of the program are important: the project must be carefully picked for size and suitability, and the mentor must be someone who can delegate tasks to the intern effectively. As long as the tasks you ask the intern to help with are

educational (and not mindless busy-work), this situation will be beneficial to all. You, the harried system administrator, get someone to help you, though there will be a certain amount of time lost to training – remember, this is a learning experience. The student gets valuable resume fodder and a foot in the door with your company. Best of all, you have a ready-made pool of talent to choose from as the interns graduate and begin looking for jobs. You know them already and know that they will fit into your group, and that they will hit the ground running in their new full-time job.

- Get involved with local colleges.

    Even a "rich" private school has budget problems when faced with the fast pace at which computing and network hardware is changing. What percent of colleges do you suppose have started using Network Appliance fileservers or Fast Ethernet? New hires can't have the experience you hope for unless they have had the equipment on which to learn.

    Find out what kind of corporate giving programs your company has available and take an active part in supplying technology to the computer science department of a local university. Consider getting your company to write off old hardware instead of trading it in with the vendor. Once again, everyone wins: local students get experience on technology more like what they'll face at your site, and you get recruits with experience which is valuable to you.

So, there you have it. Four perfectly simple things you can do to make the world better for students trying to get jobs in the system administration industry. And why should you do something like that? Because it makes your hiring process more streamlined and decreases the odds you'll make a costly mistake when hiring a student. Let's face it, in some cases, you are under more pressure to find a good employee than the student is to find a job. It helps everyone to get the inside track on up-and-coming students through these methods.

You'Goo

by Tina Darmohray
<tmd@iwi.iwi.com>

"UNIX Guru Universe; Dedicated to all UNIX System Administrators that are underpaid, understaffed, work long hours, not to mention constantly used and abused by both management and users."

Although we don't want to advertise every Web site on the Net, this one is truly special for our user community. The URL is: *http://www.polaris.net/ugu.* Thanks to Kirk Wain-grow for the pointer and the site.

USENIX and SAGE Welcome Pencom/PSA

SAGE has a new supporting member – Pencom Systems Administration/PSA.

If you are not already familiar with PSA, they are the recognized leaders in outsourcing system administration. They are currently hiring the finest system administration teams in the industry, equipping them with the best tools, and are building an entirely new breed of company.

PSA's management says that systems administrators that join Pencom have immediate access to PSA's exclusive "Collective Intellect"™, a central repository that links all Pencom System Administrators with the top talent throughout the organization and provides the best solutions to the most obscure technical problems. System administrators thrive at PSA because they are motivated by growing a business, building their career, tackling the toughest networks, and satisfying the most discerning customers.

Visit their Web page (*http://www.pencomsi.com/*) or call 1 800 PENCOM4 to learn more about the Pencom Systems Administration/PSA.

To become a supporting member of USENIX or SAGE please contact *office@usenix.org.*

Services for SAGE Members

If you are a SAGE member with a business Web page you may wish to link it to the SAGE Web site. Check out the SAGE Services Page: *<ftp://ftp.sage.usenix.org/ pub/sage/ hypertext/member-services.html>* a "guild-hall bulletin board" where SAGE members can have pointers to their business Web pages.

Attention System Administrators

You may be interested in the "Report on X/Open Distributed Systems Management" on page 37 in the Standards section of this issue of *;login:*.

Interview with Steve Johnson

by Rob Kolstad
<kolstad@BSDI.COM>

Steve Johnson

[Editor's Note: ;login: conducts occasional interviews with industry personalities. This interview was conducted electronically with Steve Johnson, President of USENIX <scj@mathworks.com>].

Rob. Since you're stepping down from the USENIX Board of Directors in June, what do you see as the major challenges facing the organization for the next year and the next five years?

Steve. UNIX is what made us happen, and what made us great, but it is no longer the major focus of our activities. We have been at the forefront of everything from C++ to Mobile Computing to Electronic Commerce to Systems Administration to Perl, Tcl, and Java. One challenge is to keep surfing the breaking wave of technology, being willing (as we were with C++) to retire conferences in areas where the technology isn't driving things any more to make room for the new stuff.

Another challenge is to avoid becoming bureaucratic as an organization. Our current staff is lean, mean, and incredibly productive and good at what they do. This lets us run an organization of modest size without ruinous dues while providing good member services. It also really pays off when we go through a recession and manage to stay in the black, as we have just done. But organizations tend to bloat up the minute the board gets distracted – it's something in the air in the US, I think. So one challenge is to stay focused on the services, and continue to hire and support an outstanding, but small, staff.

Rob. You have been on the board, with one 2-year hiatus, since 1984. What USENIX activities have given you the most satisfaction over that period?

Steve. There are two that really stand out. One was working with Rick Adams and the board to launch UUNET. The USENET system was nearing collapse as the phone bills of big providers got exponentially larger. I think we did two things right with UUNET: starting it, and then spinning it off. Both USENIX and UUNET are much healthier separately than they would have been together.

The other was my involvement in getting SAGE started, starting while I was off the board having endless Indian meals with the organizers, and working with them and the USENIX board to come up with a structure that has allowed SAGE to concentrate on the issues of systems administration and to let USENIX handle the details. In return, we have expanded our membership base and nurtured a new generation of leadership that will accomplish great things in the industry and within USENIX.

Rob. You have held many interesting industry positions over the last few years. In this era of tremendous technological hype, what do you see as the "big-deal" technologies or products over the next few years?

Steve. I spent six months or so losing money in pen computing technology, and I'm still a believer. We just think that moving a mouse so that a cursor that is a foot away from it moves is a natural act. Pointing or drawing with a pen is a truly

natural act, and someday (one or ten years from now) we will wonder how we ever tolerated using mice.

On the language area, I think the trend is towards the death of syntax as an issue – interactive editors, browsers, and builders, not to mention wizards, have already made visual X, for all X, look more similar than different. I think the trend will go away from "learning a language" to where it should have been for the last decade – learning to think about and manage the basic semantic and technological issues. In this regard, I am also a big fan of constraints – geometric and otherwise – as a way of controlling very complicated processes locally in a way that scales well globally.

Rob. What has been the biggest technology surprise for you across the last 20 years?

Steve. I'm amazed almost every month by something! The fact that computers are smaller than a house. The fact that UNIX and C have had the impact they have. The fact every Tom, Dick, and Harry has a Web page. Most of the things that amaze me aren't technological, so much, as that some of the technology wins while other technology loses, and who knows why? Why X instead of NeWS? Why C instead of Bliss or Algol 68? Why are there 25 different versions of UNIX, making all of them much less than the sum of their parts?

Rob. The portable C compiler was so popular for so long and was, by itself, a sort of de facto C compiler standard. There are plenty of C (and C++) compilers these days, too often offering features that don't interoperate. Perl, on the other hand, holds the position of sole-source for its technology. What do you think about the evolution of the compiler market?

Steve. Perl has been a wonderful example of a language that, to my taste, has little to recommend it except its indispensability! Quietly, without fanfare or any significant commercial push, it took over the world, because it offers in practice what C now offers only in theory – a uniform interface to UNIX. The 5000-line shell file that builds Perl on any known UNIX system is as much a wonder as anything else in the language and a large part of its success.

From a commercial point of view, it is very hard to make money in the language market these days. There are free products like Linux and gcc that are of surprisingly high quality, and Perl and Tcl/Tk are essentially free. There is some money to be made in the tools market for a while, but that is getting squeezed, too.

The irony is that I think our software productivity has not begun to keep pace with the drop in hardware costs. We are taking techniques of hand-carving marble and applying

them to carving butter. And soon we will be carving water. How many programmers can the world support in 20 years if we keep doing it this way – maybe roughly the number of wood-carvers it supports now?

Rob. Some of the companies you joined were "startup" companies, with smaller staffs and smaller capitalization than, say, Microsoft. Do you believe such companies can compete in the long-term or will behemoths become the de facto rulers of the computer industry?

Steve. I think it all comes down to leadership. Microsoft is well-led. Most of the well-led big companies were once well-led small companies. What does surprise me is that some large companies have been far off the price/performance curve for many years before reality intruded. In a small company, big mistakes are usually quickly fatal.

Even Microsoft is a net sink of ideas – I think the majority of the ideas in Microsoft products were developed elsewhere. Microsoft had the will and the wisdom to recognize their value and (don't ask how) get them into their products. Many other companies, frequently including the places where those ideas were developed, failed to see the application of these same ideas.

Why has UNIX always been a pig to administer? Why did faxes take over the world when email had been around for 15 years? Why did the UNIX market, which pioneered computer games, sit back and let several inferior technologies make billions from it? Leadership, or lack thereof...

Rob. You moved to the San Francisco Bay area with a family and two fabulous pianos. Now that you've had several years to enjoy Northern California, do you see it continuing as the technological mecca or are alternative communities and telecommuting going to erode the technical population?

Steve. I love it here! Companies here are like large ships floating in a sea of infrastructure – everything from Fry's Electronics and second hand office furniture stores to venture capitalists and Stanford and Berkeley. And always that Brownian motion of people hopping from ship to ship and carrying ideas, but, more than that, attitudes.

I've tried telecommuting, and loved various parts of it, but I don't think it will ever replace shmoozing. Companies have a culture, and good companies have a rich culture of aggressive enjoyment of the activity of creation, and celebration of the joy of delivering solutions to worthy customers. That's hard to create remotely, and even harder to enjoy remotely.

But I sure am productive as a programmer sitting in my home office!

Rob. What educational experiences best prepared you for the fast-changing computer world?

Steve. I never studied any computer science. I was a math major with more than a little physics on the side. It was valuable to understand the physicists' view of math – they see it as a mere tool to be thrown at the really interesting problems, no matter how much I might have felt that the math itself was interesting.

That's the same kind of attitude that most of the world takes towards software. Kind of like axle grease – you need to smear some of this messy stuff around to keep things from freezing up, but try not to touch it without gloves and wash your hands afterward. It can be a bit hard on your ego to have to deal with this attitude, but ultimately it is very focusing.

As an aside, in my perfect world there would not be much undergrad computer science taught – people should learn to do something real first, and then study CS in graduate school. Sort of the way that library science is taught. We might attract a different kind of person to CS in this case, but I think the software world would be better off with these kinds of people writing software.

I am continually surprised at how important the non-intellectual component of CS is, especially in software. Understanding what to do is often harder than doing it. Using someone else's program is harder than rewriting it. Running a project or defining an interface or deciding on a GUI can be exhausting and frustrating, especially if you approach it the same way as writing a program.

I went to a small, liberal-arts college (Haverford), where I took English classes with people who became newspaper editors, physics with future Nobel Prize winners, and they took math with me. I came away with a lot of respect for the number of different ways people's minds can be wired up, and how we all have a "place in the choir." It's a miracle when I find people who love to do the things I hate to do, but it happens a lot if you look for it.

As a result, I often find large companies much too homogeneous for my taste. At various times in my career, I have hired russian history majors, poetry majors, and one woman who had run a winery for the previous few years. Of course, these people had qualifications for the jobs I needed them to do. But they also had the mind-wiring that made them do these jobs extremely well. And most large companies would have never hired them at all, or would have paid them a tiny fraction of what they were worth.

The Web Master

by Dave Taylor
<taylor@intuitive.com>

Last issue I introduced you to the intricacies and capabilities of the Common Gateway Interface, the environment within which you write programs to supplement the capabilities of a Web server. Just about any program that emits output can – with a simple two line prefix – also produce Web pages on your server. Further, it turns out that when the browser program connects with a server, it sends a variety of information about both itself and its connection to the server, through a set of environment variables.

This time I'm going to explore some of the capabilities that this set of variables offers, then step into some very simple forms and processing scripts to see how you can ask the visitor to your site for information, then process it and respond with what's hopefully a context-sensitive Web page. The examples I'll use are a remote `ping` and `finger` capability.

How Fast Is Your Connection?

First off, recall that when a browser connects to a server, the hostname of the client program is sent as the environment variable `REMOTE_HOST`. Knowing that, it's straightforward to write a shell script to run on a server and indicate the speed of the connection between the visitor and the server system:

```
#!/bin/sh -f
echo "Content-type: text/html"
echo ""
echo "<HTML>"
echo "<H1>Ping info to host
$REMOTE_HOST:</h1>"
echo "<BLOCKQUOTE><PRE>"
ping -c 10 $REMOTE_HOST
echo "</HTML>"
exit 0
```

If you connected to this script via the Web, here's what you might see:

```
Ping info to host test.intuitive.com:
PING test.intuitive.com
    (205.149.165.109): 56 data bytes
64 bytes from 205.149.165.109: icmp_seq=0
    ttl=47 time=351 ms
64 bytes from 205.149.165.109: icmp_seq=1
    ttl=47 time=286 ms
64 bytes from 205.149.165.109: icmp_seq=2
    ttl=47 time=310 ms
64 bytes from 205.149.165.109: icmp_seq=3
    ttl=47 time=293 ms
64 bytes from 205.149.165.109: icmp_seq=4
    ttl=47 time=291 ms
64 bytes from 205.149.165.109: icmp_seq=5
    ttl=47 time=293 ms
```

```
64 bytes from 205.149.165.109: icmp_seq=6
    ttl=47 time=302 ms
64 bytes from 205.149.165.109: icmp_seq=7
    ttl=47 time=284 ms
64 bytes from 205.149.165.109: icmp_seq=8
    ttl=47 time=289 ms
64 bytes from 205.149.165.109: icmp_seq=9
    ttl=47 time=297 ms
--- test.intuitive.com ping statistics ---
10 packets transmitted, 10 packets
received, 0% packet loss
round-trip min/avg/max = 284/299/351 ms
```

Because `ping` computes an average round-trip time, in milliseconds, you could even add to the script some smarts that would let you make different decisions based on the speed of the connection between the server and the machine that's connected:

```
ping -c 10 $REMOTE_HOST > /tmp/pingme.$$
average="'tail -1 /tmp/pingme.$$ | awk -F/
    '{ print $4 }''"
```

Now "average" has the average ping speed, in milliseconds, which you can then use as a gauge to deliver different information based on connection speed. Here's a very simple example of different graphics:

```
if [ $average -lt 100 ] ; then
  echo "<img src=hi-rez.jpg>"
elif [ $average -lt 200 ] ; then
  echo "<img src=low-rez.jpg>"
else
  echo "<img src=black+white.gif>"
fi
```

Here, if you connect over a slow line, you'll get the black and white GIF format; but if you connect to the same site over a really fast line, you'll be surprised to see a high-resolution JPEG image.

Remote Ping

But what if you want to specify what machine you'd like to have pinged? It turns out that there's another environment variable that can be sent from the browser to the server, called QUERY_STRING. If you specify the URL of a particular CGI program or script and add a "?," anything following that question mark is sent as the value of the QUERY_STRING variable.

Here's how the "ping" script could be modified to check the connection between the server and any arbitrary system on the Internet:

```
#!/bin/sh -f
echo "Content-type: text/html
echo ""
echo "<HTML>"
```

```
if [ "$QUERY_STRING" = ""] ; then
  echo "<h1><i>no query string? No host to
check</i>
    </h1>"
else
  echo "<H1>Ping info to host
$QUERY_STRING:
    </h1>"
  echo "<blockquote><pre>"
  ping -c 10 $QUERY_STRING
  echo "</pre></blockquote>"

fi
echo "</HTML>"
exit 0
```

So let's say that the previous script is called "query.sh," and that it lives on "test.intuitive.com" in directory "cgi-bin." The URL for that particular script, then, is *http://test. intuitive.com/cgi-bin/query.sh.* But now I can actually send it some arguments – in this case the hostname of a machine to check – by simply appending that to the URL itself. So within, say, Netscape Navigator, I can click the "open URL" button and type in *http://test.intuitive.com/cgi-bin/ query.sh?pipeline.com*

Here's what would be shown as the contents of that page:

```
Ping info to host pipeline.com:
PING pipeline.com (198.80.32.3): 56 data
    bytes
64 bytes from 198.80.32.3: icmp_seq=0
    ttl=244 time=39 ms
64 bytes from 198.80.32.3: icmp_seq=1
    ttl=244 time=36 ms
64 bytes from 198.80.32.3: icmp_seq=2
    ttl=244 time=41 ms
64 bytes from 198.80.32.3: icmp_seq=3
    ttl=244 time=42 ms
64 bytes from 198.80.32.3: icmp_seq=4
    ttl=244 time=29 ms
64 bytes from 198.80.32.3: icmp_seq=5
    ttl=244 time=32 ms
64 bytes from 198.80.32.3: icmp_seq=6
    ttl=244 time=31 ms
64 bytes from 198.80.32.3: icmp_seq=7
    ttl=244 time=30 ms
64 bytes from 198.80.32.3: icmp_seq=8
    ttl=244 time=32 ms
64 bytes from 198.80.32.3: icmp_seq=9
    ttl=244 time=31 ms
--- pipeline.com ping statistics ---
10 packets transmitted, 10 packets
    received, 0% packet loss
round-trip min/avg/max = 29/34/42 ms
```

As you can see, Pipeline in New York has a much, much faster TCP packet turnaround than my workstation!

A Form Front-end

Being able to add arguments to a URL as a way to pass information is useful, but it's somewhat limited, as you might expect. After all, what you really want to have on your Web pages are boxes and checklists, places where users can specify information then press a "do it" or "submit" button and have that information quickly relayed to the waiting CGI script.

That's done by using the `<FORM>` HTML tag set within your documents, as shown:

```
<HTML>
<h1>See how fast we're connected to the
        net!</h1>
<hr>
<FORM METHOD=get
ACTION="http://test.intuitive.com/cgi/
        query2.sh">
Look for?
<input type=string name=ping>
<P>
<input type=submit value="ping this host">
</form>
</HTML>
```

Most of this is hopefully straightforward Web markup, until you get to the `<FORM>` section. The form tag has two attributes: the mechanism by which the information should be transmitted to the server and the URL of the CGI program that should receive the information. In this case, you can see that the `ACTION` specifies that the remote script is referenced as *http://test.intuitive.com/cgi/query2.sh* and that the `METHOD` is `get`.

There are two basic `METHOD`s for sending information from a browser to a CGI script: `get` and `post`. `get` is easier to work with because the information is all tucked neatly into the `QUERY_STRING` environment variable, but it has some serious size limitations. Instead, complex Web forms invariably use `post`, which sends all the information as standard input to the CGI script, enabling an arbitrary amount of data to flow from the browser to the CGI script. I'll explore `METHOD=post` in my next column.

Every input field from an HTML form is sent as a `name=value` pair, with multiple pairs separated by an ampersand. In the case of this particular form, the `<input type=string>` HTML tag produces a small box within which the user can type the name of a system to ping. That information is actually sent to the CGI script as "ping=whatever-they-typed" (that's what the `name=ping` does in that `<input>` tag). There's a variety of different input type fields, including those shown in the following table:

input type	meaning
string	one line of text requested from the browser
password	one line of text - not echoed as typed
radio	one of a set of "radio" buttons
checkbox	a yes/no checkbox
submit	the 'submit' or do it button
reset	the 'reset' or restore default values button

There are lots of good references to this material both online and in books available at your local bookstore, so I won't belabor the point here.

Now I have a simple Web page that's going to prompt for a host to ping and right below it present an action button that the user can press to have her information sent to the script and acted upon. That variation of the CGI script looks surprisingly similar to the last version we saw. Indeed, all I need to add is the ability to decouple the hostname from its `name=value` form:

```
#!/bin/sh -f
# modified to accept name=value pairs...
echo "Content-type: text/html"
echo ""
echo "<HTML>" if [ "$QUERY_STRING" = ""] ;
        then
    echo "<h1><I>no query string? No host to
        check</i></h1>"
else
    host="'echo $QUERY_STRING | awk
        -F= '{print $2}''"
    echo "<H1>Ping Info to Host $host:</h1>"
    echo "<blockquote><PRE>"
    ping -c 10 $host
    echo "</pre></blockquote>"
fi
echo "</HTML>"
exit 0
```

I make use of the `awk` program to split the information received at the `=`, which works fine for a single value, but if we move to a multiple variable script, a more sophisticated technique will be required.

With this HTML page and the script shown above, I can now pop over to my Web site and ping any host I can think of on the network with wild and merry abandon. Entering `pipe-line.com` and pressing the submit button would even produce results identical to those shown earlier.

Another Example: Finger

Let's see how this same technique can be applied to another simple UNIX command, one that would be useful to access

from within the Web environment: the `finger` command. It's an interesting command because its behavior depends on the type of information you give it: use it to `finger` a name and it will search for everyone with that information in the password file, showing you all the results. Give it a remote hostname instead – in the form `@hostname` – and it will tell you who is logged in to that machine. Use a fully qualified email address – `user@hostname` – and it'll tell you about that person if it can connect to the machine.

Here's the HTML file I created as a quick and simple `finger` front-end. You'll see it's remarkably similar to the `ping` page:

```
<HTML>
<h1>Finger: find out about users or
  computer systems!</h1>
<hr>
<form method=get action="http://test.
  intuitive.com/cgi-bin/finger.sh">
Look for?
<input type=string name=finger>
<P>
<blockquote><font size=+1>
<I>Try an email address for a specific
  user, or just the '@hostname' format
  to see who is using a particular
  computer on the net</i>
</font></blockquote><P>
<CENTER><input type=submit value=
  "look up this user or host">
</CENTER>
</form>
</html>
```

This time we've also included some helpful information for the user in the form of a brief italicized comment below the input box.

The script at the other end looks like this:

```
#!/bin/sh -f
# Finger user or user@host or @host
echo "Content-type: text/html"
echo ""
echo "<HTML>"
if [ "$QUERY_STRING" = ""] ; then
echo "<h1><I>no user or host to
check?</i></h1>"
else
value="'echo $QUERY_STRING | awk
-F= '{print $2}''"
echo "<H1>Finger information for
$value:</h1>"
echo "<blockquote><PRE>"
finger $value
echo "</pre></blockquote>"
fi
echo "</HTML>"
exit 0
```

In the spirit of good coding, we again include some error checking code – this time the CGI script can produce an error message (as proper HTML, of course) if you hit the script without having gone through the HTML page (if you leave the box blank, it sends `finger=` so there's still some data). If you do give it something, it will show you the results of the `finger` command run on the server with the information you've specified.

Let's have a quick look at some of the possible output formats. Note in these examples that the `$value` sent in also appears as part of the output. As an interface rule, this is a great bit of positive user feedback, enabling users to verify that what they sent is what was processed.

First off, how about what happens if I don't specify anything?

```
Finger information for:
  Login       Name            TTY       Idle         When          Site Info
  taylor      Dave Taylor     p0        2            Thu 12:57
```

Or, if I specify `@usenix.org` to see who might be logged in there:

```
Finger information for @usenix.org:

[usenix.org]
Login            Name              TTY     Idle    When         Where
zanna            Zanna Knight      KA      1:46    Thu 09:00    Zanna's Mac:8.23
toni             Toni Veglia       co      13:     Tue 14:59
diane            Diane DeMartini   p6      8       Thu 08:44    131.106.3.16:0.0
ellie            Ellie Young       p7      1:22    Thu 09:04    boss:0.0
toni             Toni Veglia       pb              Thu 09:12    131.106.3.17:0.0
carolyn          Carolyn Carr      pe      1:26    Thu 09:20    bigx:0.0
ah               Alain Henon       q0      12      Thu 09:29    131.106.3.29
zanna            Zanna Knight      q3      59      Thu 09:48    131.106.3.20:0.0
eileen           Eileen Curtis     q5      36      Thu 10:04    131.106.3.19:0.0
```

Lots of people and lots of things going on at USENIX, as always! Of course, I can also pick someone and submit their name@usenix.org to find out more about them.

```
    Finger information for zanna@usenix.org:
[usenix.org]
Login name: zanna          In real life: Zanna Knight
Directory: /staff/zanna      Shell: /bin/csh
On since Dec 7 09:00:41 on KA0.0 from Zanna's Mac:8.23
1 hour 45 minutes Idle Time
No unread mail
No Plan.
Login name: zanna          In real life: Zanna Knight
Directory: /staff/zanna      Shell: /bin/csh
On since Dec 7 09:48:06 on ttyq3 from 131.106.3.20:0.0
58 minutes Idle Time
```

Here Zanna is actually logged in on two different lines, which is why we see two entries for her in this output. Notice that the top entry indicates that she's actually connected from a Macintosh too. Ah, the things you can glean when you poke around on the network . . . .

Enough . . . .

We've probably gotten into enough trouble for this issue of *;login:*. Next time, as I said, we'll look a bit more closely at how variables are sent from the browser to the server, including figuring out how to decode all the weird encoding formats used to ensure that unusual characters travel across the wire without any major problems.

Musings

by Rik Farrow
<rik@spirit.com>

It's been about a year since I started writing for *;login:*, and it's been very unusual for me. Wonderful editors, although as pushy as ever about deadlines, and no assignments. The letters I get back, in the form of email, are informative, not rants – a real difference from one of my previous incarnations (technical editor of *UNIX World* magazine).

I never worked for the magazine. I was a contractor and got involved by volunteering. I had been invited to sit in on the yearly editorial meeting, and at the end of it, I caught up with the assistant editor and told her that the magazine often contained glaring technical errors, some that even the most inexperienced UNIX user would recognize. Shouldn't they have someone who read through the articles looking for stuff like this?

She turned it right around and asked me if I would do it. I said, sure, if you'll pay my normal consulting rate. I thought that would put an end to that quickly, but I was wrong. That's how I wound up reading a magazine cover to cover, sometimes twice, and getting paid for it.

As time went on, I got more involved. I began to understand the basic contradiction inherent in most computer magazines, some more than others. The people putting out the magazine have little to do with computers. The editors are English majors, maybe there's one or two journalists, then there are salespersons, managers, artists, etc. The new-products editor was particularly creative when it came to translating computer terms. Least significant bits would become "the part of the number that doesn't mean much to people" under his clever hands.

I always worked to steer the magazine toward more technical articles, but as time went on, I had less and less success. I found myself sitting, at my remote office, conference-calling into the weekly meeting and at the same time practicing my marksmanship with an air pistol. I had no idea how angry I was about the whole thing until I took off for a couple weeks and decided to part ways. Not only did I feel better, but when I got back home, I got a phone call from the editor telling me they had reached a similar decision. Actually, they had decided to become a nontechnical computer magazine and didn't need me anymore.

That was about a year ago. I just bought the final issue of *Open Computing*, the magazine's new name, in a used book store. Not that I wanted to read it. I just wanted it for my collection: maybe it will be worth something some day. There are two morals hidden in these ramblings: 1) Volun-

teering can open some interesting doors, and 2) It's better to leave than try to drag along a whole gang intent on heading in another direction, no matter how stupid it seems to you.

The Meat or the Beef

Not that magazine editors have any corner on the ridiculous. In an article in *ComputerWorld's Client/Server Journal* (November 1995), Ram Sudama stated, "I believe it's inevitable that [DCE] will succeed." Sudama, of course, is somewhat biased. He is vice president of Open Environment Corporation, a developer of tools for DCE application builders. He was also largely responsible for the Open Software Foundation's (OSF) Distributed Computing Environment (DCE).

A couple of years ago, I started asking participants in my security classes if their organizations were using DCE. I believe I started this after hearing Rich Stevens ask the same question. The results have been remarkably consistent – from zero to two hands go up, representing perhaps 2% on the average. I wanted to know if I needed to deal with DCE in my class materials. So far, DCE appears in the pilot project stage.

DCE is an interesting idea, one at the foundation of client/server. DCE is supposed to turn an entire network of heterogeneous computers into a single, smoothly functional unit. Hypothetically, this is a thing of beauty. Practically speaking, it has turned into a nightmare.

To use DCE, you first organize a group of computers into cells. Each cell must support at least one directory server, a security server, and several time servers. There may also be other servers that use these basic services and provide other services, such as file or database serving. The idea is that DCE becomes a new layer on top of the existing operating systems that appears identical everywhere.

So far, everything sounds okay. But here's the meat of the matter. You must administer this DCE cell and each of its services as separate from the individual systems. You have not only added a new layer of abstraction over your system: you've added a new layer of incompatible system management to boot. And that's my beef.

When I saw what a mess it was even to begin administering a single DCE cell, I decided I had wasted enough time. If you have gone this route, you know that you've had the opportunity to learn the ISO naming conventions (as the rest of us might have seen in X.400 email). DCE can interoperate with DNS, but goes well beyond, thus missing one chance for reducing the mess created by another level of administration.

DME, the Distributed Management Environment, was supposed to ease the pain of administering DCE cells. As it stands, DCE supports no form of centralized administration. Everything is supposed to be distributed and replicated for reliability. Hello there, OSF. PC desktop managers and the companies selling them operating systems have been desperately trying to centralize administration, instead of scattering it. And OSF gave up on DME several years ago.

I have been on the lookout for DCE in action – a success story not related to a pilot program, something real life. But in vain. I almost got to see a demo of Transarc's DCE-based transaction monitor at UNIX Expo a couple of years ago. I kept going by the Transarc booth, but they could never get all the DCE daemons running at the same time, so the demo didn't happen. I was impressed. Here were developers with a product based on DCE, and they couldn't get it to work reliably. Not that a show floor is a hospitable place, but still.

Enter Java

I think that much lighter-weight ventures may succeed where the extremely heavyweight DCE has failed. Take Sun's Java as an example. It supports the distribution of applets across a network. The applets are architecture neutral and can be run securely (depending on the local environment). Instead of relying on security servers, public key signatures are used for authentication.

What's different here? No massive infrastructure is required to support Java. There is certainly a missing piece (the method for distributing public keys), but the design is simpler and more elegant. Not only does Java support client/server, it also supports its own distribution, solving another administration problem. It does *not* add a new, completely incompatible, layer of system and network administration.

I don't want to learn yet another complex administrative scheme. I don't want to learn another way of addressing systems and processes on those systems. Somehow, I always suspected that OSF wanted to take over the world, and they would do so by making us do things their way. Perhaps I'm a bit paranoid (I am, but that's what security people get paid for), and that's what DCE looked like when I became aware of the administrative nightmare waiting in the wings.

Perhaps you can share a DCE success story with me. I'm still waiting. I'm also looking for stories about successes using commercial system administration software for distributed, heterogeneous networks. Let me know if you're out there.

The Brazilian National Program of Open Systems Reference Centers

by Fabio Queda Bueno da Silva
<fabio@di.ufpe.br>
and Silvio Lemos Meira
<srlm@di.ufpe.br>

Introduction

The Brazilian Reference Centers Program, named CENAR, was developed by the Brazilian National Research Council, CNPq, (a funding agency), through its national program ProTeM-CC (the Multi-institutional Program in Computer Science). The main objective is to establish some centers where the issues and technologies related to heterogeneous computing environments, their installation, operation, and administration are discussed.

Historical Background

ProTeM-CC is a special action of the Brazilian Federal Government under execution of the Ministry of Science and Technology and CNPq. It is one of the three official programs in computer science regarded as high priority by the Brazilian Government. The two other programs are the National Research Network (RNP), the Brazilian access to Internet, and Softex-2000, the Brazilian effort to export software.

The ProTeM-CC program started when an action to equip and update Brazilian computer science departments' research laboratories was launched in late 1990. Around half a year later, about 230 SUN Sparc stations were being installed in laboratories that had until then been populated by aging XT and 286/386 machines. The program's other actions, through scholarships and direct funding, got under way in early 1992 and have already funded US$9 million worth of research.

A tender to buy further computing and subsidiary equipment was issued in May 1994, and the process is now complete. As a result, the current number of seats in research laboratories has doubled since May 1995. Furthermore, 21 collaborative projects, involving over 50 institutions, both private and public, are being directly funded in the last months. This will further increase the size and quality of the academic and research laboratories in computer science in Brazil, through equipment acquisition carried out directly by the projects.

During the last couple of years, the national coordination of the ProTeM-CC program has identified two major problems in trying to plan future expansions of the laboratories of its associated institutions. On the one hand, to define the acquisitions of large amounts of equipment (as in 1990 and 1994) is a very difficult and demanding task that requires specialized personnel actively involved with test and evaluation of new technologies of hardware and software, from an array of suppliers, in a fast evolving market. Professionals with such profiles are not easily found either in academic or industrial sites.

On the other hand, ever since the first lot of equipment was acquired by ProTeM-CC, many laboratories around the country, for one reason or another, were purchasing different hardware and software and creating heterogeneous environments that, although always necessary due to the particulars to which certain makes of hardware and basic software are suited, created an enormous set of problems.

In order to look for solutions to these problems, ProTeM-CC has decided to establish some centers where the issues and technologies related to heterogeneous computing environments will be discussed. Such "national open systems centers" (Centros Nacionais de Referencia em Sistemas Abertos – CENAR, in Portuguese) are thought to be natural joint enterprises among hardware and software suppliers, research centers, and key users of advanced networked computing technology.

The Reference Centers Program

Professional system administration techniques and tools and a well-qualified team of system administrators are essential to the smooth running of large heterogeneous installations. The costs of achieving high quality in the administration of such installations are, not surprisingly, very high, thus making the sharing of experience, tools, and expertise among academic sites an essential item. To make sharing possible, unified and consistent models of systems administration are required. These unified models would allow sharing of existing solutions and an optimization of the efforts necessary to maintain the installations in the academic and nonacademic sites.

However, there are wide differences between the models of system administration used in academic networks. Most sites have established systems that could not readily be made more unified and efficient. This leads to the idea of establishing a number of nationally distributed reference centers that would be able to experiment with new system administration techniques, define reference models of system administration, and gradually move those techniques and models into surrounding sites.

The centers would also train professionals in the administration of heterogeneous computing systems. Such professionals are in high demand in the industrial and business sector, and very few specialist courses covering the large range of skills necessary for this task are available.

Furthermore, the centers would carry out tests, evaluations, and comparisons among different computing platforms with respect to their performance and interoperability in a heterogeneous network. ProTeM-CC would use the results of such evaluations, in the form of reports, to define the basis for future equipment acquisition. ProTeM-CC funded projects would also be instructed to obtain guidelines for equipment acquisition from the reference centers.

Therefore, the establishment of reference centers throughout the country would help in optimizing the use of existing computer resources and in the strategic planning of future expansions of the academic networks at the national level by defining models of system administration and training programs that will make better use of available hardware and software; training system administrators in the tasks of managing the operation and expansion of large heterogeneous computing installations based on open systems; and studying, testing, evaluating, and certifying new technologies and alternative solutions that will guide future expansions of existing installations.

The reference centers would also benefit from developments in the field of (wide area) network management carried out by the National Research Network Program (RNP) and complement their effort with results on the administration of heterogeneous local area networks. The Softex-2000 program would also be a natural partner in the reference center effort, playing the role of a client that can pose challenging problems to the centers and as a source of expertise that can also be used in their solution.

With the above ideas in mind, the ProTeM-CC program and CNPq have started a special action with the objective of installing and supporting the operation of some reference centers in Brazil. A common structure underlining the actions of these centers was established to coordinate the developments and avoid duplication of efforts and consequent waste of resources. This structure is the subject of the rest of this article.

National Structure

Some guidelines rule the establishment of reference centers. The centers must be distributed around the country and located in different regions to allow better geographical coverage. This is particularly important if the centers are to provide support to surrounding sites, specially in a large country like Brazil.

Each reference center must be closely attached to and under the coordination of a local university or research center with excellence in research in areas such as computer networks, distributed systems, and system administration.

The goals and directions of the centers' operations are defined at the national level by a board constituted of representatives from the industrial sector, the government, and academia. National coordination ensures that the board's common objectives are being followed at each individual centre. This national coordination is situated at the ProTeM-CC headquarters, in Recife.

Local Structure

At the local level, each reference center carries on research and development on heterogeneous systems administration, studying and developing methods and tools for the administration of large networks of workstations with heterogeneous platforms of hardware and software. Software and hardware products are evaluated and compared with the objective of guiding future expansions on the networks. The reference center produces reports on new technologies of hardware and software, tests and compares alternative solutions, and is in contact with industry to learn of and possibly develop new products related to its mission.

The reference center provides reference models for network administration that can be used in installations throughout the country, providing support for sites with different levels of organization and size. As part of this activity, the center also provides software and information archives for system administration that document the reference models.

The activities developed at each center should help enhance communication with surrounding sites by providing training in system administration and related topics and assistance on the use of reference models of system administration.

To achieve these objectives, a two-level structure organizes a reference center at the local level. On the operational level, the center's activities are driven by processes, i.e., projects to look for solutions to specific problems, each of them executed by a task force. Each task force carries out research and development activities, prepares and delivers training programs and defines and implements strategies to externally market the products of its activities. Once it is assigned a process, a task force works as an autonomous enterprise unit. Processes are defined either at the national level, in which case they are developed at one or more reference centers, or at the local level to cater to regional or local demand.

At the administrative level, local coordination is responsible for all management aspects of the center, including personnel and resource allocation to processes, technical supervision of the research and development activities of the task forces, and the administration of the center. The coordination of each center is also responsible for keeping the center's activities on line with the main national goals of the reference center program. This management structure is very flexible, allowing personnel and other resources to be relocated between processes according to necessity.

A reference center is sought to be a small enterprise of at most 20 technical staff and very few administrative personnel. The desirable size of each task force is between four and six people. As the number of technical staff starts to increase beyond these optimal limits, groups may be motivated to spin off from the center as private enterprises. In this way, the center will also contribute to the generation of advanced technology-based startups in the local community.

The local coordination supervises the activities of the task forces at three levels: (1) research and development activities related to research on heterogeneous system administration, evaluation of software and hardware technologies and products, and definition of reference models for network administration; (2) support activities related to the definition of training programs and support schemes to be delivered to the client sites and the installation and maintenance of software and information archives; and (3) external communication activities related to the establishment of communication with institutions outside the reference center, mainly local software and hardware businesses, to promote links and exchange know-how in system administration and related areas.

Research and Development

Most of the research and development carried out by the task forces are toward the implementation and evaluation of models of systems administration with the objective of establishing general reference models. To develop these reference models, the following issues are observed: management efficiency, resilience, predictability, transparent upgrading and reconfiguration, consistent and stable user view, and security.

The very nature of the research work carried out at the reference centers requires a computing laboratory detached from the host institution's main laboratory. This allows experiments to be carried out without disrupting the services to the general users.

Each task force is responsible for setting up and maintaining its own research and development laboratory using a pool of equipment composed of heterogeneous platforms of workstations, personal computers, routers, bridges, hubs, etc.,

supplied mainly by the manufacturers that are the partners of the program.

The research activities are carried out in close connection with the research activities of the host institution. Researchers and students of the host institution also participate in the center's activities.

Support

Support activities are related to defining and delivering training programs and setting up information and program archives to be shared among the reference centers and the surrounding institutions.

Four generic training programs have been defined with complementary goals: (1) to train system administrators in the principles of management of network of computers; (2) to provide training on the reference models, their setup and operation, directed at experienced system administrators; (3) a program directed at the users of networks, to allow better use of available resources and services by providing knowledge of the basic principles behind the operation of an open systems-based network; and (4) an industry program, that should be the applied (real world) version of those aforementioned, with an emphasis on industrial applications of network technology and management techniques in the industrial sites.

Specific training programs can be defined by each task force, related to their own research and development activities. In this case, the task force is responsible for setting up training facilities using the pool of equipment. A lecture room with multimedia facilities is shared between generic and specific training programs.

For the installation and running of information and software archives, the task forces can use a dedicated fileserver. Those archives keep software packages that implement the reference models, articles, technical information on equipment, research reports, etc.

External Communication

Each task force is responsible for increasing the awareness of the importance of system administration within the institution (internal marketing) and to promote the reference center in the surrounding academic and industrial community (external marketing), by means of seminars, visits, advertising, etc.

In addition, a showcase is organized with the following objectives: to show the development of the reference centers, to show the usage of an infrastructure of hardware and software to solve some common network problems, and to show the state-of-the-art of Brazilian system administration.

Further Information and Contact

For further information about the reference center program, please contact the national coordination of the ProTeM-CC program at the following address:

ProTeM-CC –National Headquarters
Departamento de Informática - UFPE
CP 7851 - Recife - PE - CEP 50732-970
Brazil

Phone: +55 81 2714281
FAX: +55 81 2714925
Email: *cenar@di.ufpe.br*

Industry Initiatives for Science and Math Education (IISME)

by Kaye Storm
<kayestorm@aol.com>

In the past dozen years, a flood of state, regional, and national reports has raised the alarm about the condition of mathematics and science education in this country. Technological innovations have created major changes in the nature of the education and job training needed to produce a scientifically literate populace with a work force suited to the demands of international competition. In response, there has been an increasing emphasis on involving the resources and expertise of the private sector in addressing these challenges. From the seminal report *A Nation at Risk* in 1983 to President Clinton's recent *Goals 2000* initiative, the clarion call has been sounded for business to become more involved in education. And business has responded. Local and national alliances between education and the private sector are being formed at an unprecedented rate, offering the potential for major improvements in the quality of mathematics and science education.

One program that has proved tremendously successful in engaging business and industry in the education reform movement is Industry Initiatives for Science and Math Education (IISME).

Background, Purpose, and Accomplishments of IISME

IISME is a nonprofit organization founded in 1985 by a consortium of San Francisco Bay area companies and government laboratories in partnership with the Lawrence Hall of Science at the University of California, Berkeley. IISME's mission is to address the critical need for a strong, highly skilled work force in mathematics, science, and other technological fields. The industry-education partnership focuses on teachers as the primary agents for effecting meaningful change in mathematics and science education. IISME seeks to provide teachers with work experience in applied science and mathematics, as well as access to the latest technology and technically trained professionals. IISME places science, mathematics, and computer science teachers (primarily at the high school level) in mentored industry jobs for eight weeks during the summer. IISME provides year-round assistance to teachers as they strive to translate their summer experiences into updated and enriched classroom instruction.

In the first eleven years of the program, 80 corporations, universities, and government laboratories have participated in the IISME program by providing almost 850 fellowships for teachers from over 150 public and 40 private schools in the seven-county San Francisco Bay area. Through the IISME partnership, industry sponsors have contributed over $7 million to improving science and math education. In addition, the sponsors have contributed over 30,000 volunteer hours as mentors to teachers, coordinators of the program within companies, guest lecturers in classrooms, hosts for company tours, and counselors to IISME and schools. The nearly 500 teachers who have participated in the IISME Summer Fellowship Program represent 25% of all Bay area high school mathematics and science teachers. They have shared their newly gained knowledge with over 750,000 local students.

Year after year, over 90% of IISME teacher fellows rate IISME as one of the best professional development experiences available to them. Similarly, over 90% of the teachers annually report that their instruction has improved as a result of the IISME experience. Teachers consistently cite a renewed enthusiasm for teaching, better career counseling for their students, and more relevant, current curriculum and instruction as outcomes of their summer work. IISME teachers also increase their emphasis on teamwork, problem solving, communication skills, and professional work habits in their classrooms. In addition, a recently completed survey of all past participants in the program revealed that for 40% of the teachers, the IISME experience influenced their decision to remain in teaching. Nearly 70% reported that the IISME experience was a catalyst for further professional development.

Industry mentors and teacher fellows often collaborate on developing ideas for classroom transfer during the summer, and over 85% of mentors either make classroom visits or host students in industry. All IISME teachers commit to designing an action plan for transferring their summer experience to the classroom. Mentors sometimes assist in the design or implementation of their plans.

All mentors and teachers become permanent members of the IISME Academy and are invited to participate in several year-round IISME Academy activities. These meetings usually include a tour of an industry facility (research lab, manufacturing plant, etc.), a lecture or seminar on a technical topic, and time for small group discussions among teachers and industry representatives. Bay area companies have typically provided the tour, activities, and meeting facilities as an in-kind donation.The industry-based format of the IISME Academy has proven highly effective in allowing teachers continued access to industry personnel, facilities, and ideas. Such access is not available through other more traditional staff development programs offered to teachers during the academic year and represents a unique and important contribution by IISME to teachers' ongoing professional development.

The Bay area IISME public-private educational partnership has proven so successful that it has been replicated throughout the country and overseas. In 1993, IISME was recognized by the US Department of Education as a model collaborative helping the nation achieve the America 2000 national educational goals.

Expanding the Number of IISME Computer Networking Assignments

With general science and math as the primary focus of IISME, summer work experience opportunities with the industry participants have been largely in the area of applied mathematics, physics, chemistry, and computer applications. The number of applicants always far exceeds the available positions. As a result, IISME is actively seeking increased participation from Bay area organizations that can offer summer fellowships to local teachers. There is an especially strong need for summer assignments in the area of computer networking, one of the fastest growing industry segments and an area in which schools increasingly need experience as their technology programs grow.

Benefits of IISME to the Participating Industries

In addition to helping the educational system better prepare students for careers that await them in industry, industry gains some direct and immediate benefits from an IISME-type partnership with schools. In this era of corporate re-engineering whereby the traditional ways of doing tasks are scrutinized, teachers can provide significant input to the process by contributing fresh ideas and perspectives and different ways of looking at problems. Teachers are great communicators and have excellent problem-solving and facilitation skills. Their interactions with employees, many of whom have children attending schools, make employees

more aware of the issues and problems facing our educational system and the ways community members can help schools do a better job of educating their children. Over 90% of IISME mentors each year report that teachers outperformed their expectations and made significant contributions to their company.

[Editor's note: For information about volunteering and industry sponsorship, contact the Director of IISME's Special Projects, author Kaye Storm at kayestorm@aol.com.]

An Update on Standards Relevant to USENIX Members

by Nick Stoughton
USENIX Standards Report Editor
<nick@hoskyns.co.uk>

Report on SC22/WG15/USTAG

Charles Severance <crs@crs.cl.msu.edu> reports on the July 10-15, 1995, meeting in Nashua, NH

The most interesting topic at the United States Technical Advisory Group to ISO Working Group 15 (the "TAG") meeting was the discussion regarding the question as to whether or not we would support the proposal to give X/Open a category C liaison to the ISO SC22/WG15 standards committee. There had been a number of interesting hallway discussions about the topic. When we reached that point in the agenda where we were developing the US position on this issue, the chair called for discussion on the topic. There were several requests for clarification but no substantive discussion. There were several long pauses during the consideration of the issue.

As the chair moved toward taking the vote, I wondered what the other members of the group were thinking. As I prepared to cast my vote, many things raced through my mind.

First, X/Open is a very good standards-related organization. They have been a strong supporter of the POSIX standards. In my mind, X/Open has added tremendous value to the POSIX standards. By adopting POSIX and then filling in the gaps, XPG provided a specification that the computer vendors could buy and the computer users could purchase to accomplish real work. As a vendor-driven consortium, X/Open could fill these gaps much more quickly and respond to market needs in a time frame that allowed the wider spread of UNIX and open systems.

In addition to adopting POSIX standards, X/Open has financially supported people to attend POSIX. Without the support of X/Open there almost certainly would never have been POSIX standards for networking and system administration. X/Open has patiently participated in the IEEE/ISO process as a good citizen (which *does* take some patience).

When X/Open develops international standards in association with POSIX, their route to an international standard would take roughly the following steps:

- Develop and ballot the document as an X/Open document according to X/Open rules and procedures.

- Submit the document to the IEEE as part of the POSIX standards effort – attend POSIX meetings and ballot the document using the IEEE rules. Resolve the comments from the IEEE POSIX balloting group.

- Submit the document to ISO at SC22/WG15 and go through an international ballot. At this step, they must resolve comments from various countries such as Canada, Denmark, France, Germany, etc.

Eventually the document is approved as an International Standard.

The following reports are published in this column:

- SC22/WG15/USTAG

- POSIX.1 Removable Media Interfaces

- POSIX.1a: System API

- POSIX.1e/POSIX.2c: Security

- POSIX.1h: SRASS

- POSIX.1m: Checkpoint/ Restart

- X/Open Distributed Systems Management

Our Standards Report Editor, **Nick Stoughton**, welcomes dialogue between this column and you, the readers. Please send any comments you might have to *<nick@usenix.org>*.

However, the category C liaison can be viewed as the first step in bypassing the IEEE steps in this process. I can certainly see why X/Open might want to eliminate this step. Going through the IEEE will add at least a year to the time it will take to complete the process of getting one of their standards approved at the international level. There is no question that getting standards approved quickly is in the best interest of computer vendors and computer purchasers.

Given that one possible outcome of the category C liaison is that X/Open will begin to completely bypass the IEEE/POSIX process, one might expect the US WG15 TAG (dominated by members of POSIX) to vote not to support a liaison that might eventually cut them out of the flow of X/Open specifications. Of course, that would have only been one vote at WG15 and the European countries would have certainly supported X/Open, so the US would have been overruled. Still, perhaps it might have been a good symbolic gesture.

The other thoughts running through my head were wild speculations of a worst case scenario for the future. In this scenario, X/Open eventually creates virtually all of the open system specifications and sends them straight to ISO, bypassing IEEE. This thought saddens me at some level.

It seems that the only formal standards body on the planet where there is any significant user power is the IEEE. People constantly complain that membership (and balloting) in the IEEE is based on the individual and not by organization. In the IEEE, users who actually purchase systems have a real voice in the process. A vote from an individual engineer is equal to a vote from a corporation. I have also seen situations where engineers vote on a ballot based on their unbiased assessments as engineers rather than on the best interest of their organizations.

Once IEEE/POSIX is bypassed, the computer vendors will dominate all of the forums through which the standard passes. They dominate X/Open, and they have a very strong presence in ISO.

If I look back a few years, the UNIX market had a large number of vendors, and users had a large impact on UNIX related standards in forums such as USENIX and POSIX. As time passes, we are quickly approaching the situation where there will be only a small number of major vendors of open system products and they will control the standards users specify to procure those systems. Every standard they develop will quickly become an international standard with little or no user input.

One other choice I would have in forming my vote would be to wait to see which way the wind was blowing and vote with the majority. That way I would not have to make a choice.

In the instant the chair called for a vote, I decided to vote affirmative based on what I know to be true in the past rather than what might happen in the future. After my hand was up, I looked around the room, thinking it might be a close count, but every hand in the room that I saw was raised in support of the X/Open liaison.

Since the meeting, I have wondered why this vote turned out the way it did. My conclusion is that the yes vote was a referendum on the respect and trust between POSIX and X/Open that has developed over the years. This respect and trust are based both on individual people and on the overall organization. How could we do anything but support an organization that has contributed so much to POSIX over the years?

Report on POSIX.1: Removable Media Interfaces

Chuck DeBaun <debaun@fnal.fnal.gov> reports on the October 16-20, 1995, meeting in St. Petersburg, FL

The removable media group was formed to create an optional standard for a removable media resource management command line interface. However, in the search through existing standards, it was noted that such a standard could not be implemented in a strictly compliant POSIX environment. Indeed, serial media, otherwise known as tape, cannot be supported in such an environment.

Thus, as a first step, the removable media study group submitted a Project Authorization Request (PAR) to provide the missing `mtio` semantics in POSIX.1 so that serial media, a primary type of removable media, can be supported in a POSIX environment. This PAR was approved in July 1995. A proposal is currently on the table and is being used as a working document. This proposal is based on the BSD `mtio` interfaces. Draft 3 of this proposal is expected following the October meeting.

At the same time, a PAR (POSIX.2d) was accepted to provide the `mt` utility definition for POSIX.2. This will provide a command line interface to the mtio API. Work has not yet begun on this project.

A third PAR is being prepared for the actual removable media resource management command line interface specification. This work is being delayed by the need to create support for serial media before it can be started. The need to backtrack to create the `mtio` semantics has caused a marked decrease in interest and attendance, further delaying action in this area.

However, this is an important area for standardization, and I would strongly and urgently encourage willing participants to step forward!

Report on POSIX.1a: System API

Shravan Pargal <pargal@cray.com> reports on the July 10-15, 1995, meeting in Nashua, NH

The first meeting of the POSIX.1a working group under the new organization commenced with a plenary meeting of the entire system services working group to elect a working group chair. Both Jay Meyer, chair of the former POSIX.1 working group, and Joe Gwinn, chair of the former realtime working group, were nominated. Following some fine campaigning by both candidates, Jay Meyer was elected chair after a short ballot procedure.

The new system services group now consists of:

- core interfaces (POSIX.1a)

- SRASS (POSIX.1h)

- real time(POSIX.1b)

- removable media (POSIX.1k)

- transparent file access (POSIX.1f)

- resource limits (initially a part of POSIX.1)

- checkpoint/restart (also initially a part of POSIX.1)

Joe Gwinn, as runner-up in the election for chair, was appointed without contest as vice chair of the new working group.

The POSIX.1a document went through some major reorganization during this meeting. Having experienced significant difficulty in getting the checkpoint/restart and resource limits sections through the ballot process, it was decided to split this work off into two separate new projects. There is work to do on these areas, and the ballot group had alerted us to the fact that they weren't yet ready for publication. New Project Authorization Requests (PARs) were submitted to the PAR Management Committee (PMC) for these projects. The PMC agreed to recommend approval of the two new PARs. It was decided that separate ballot groups, separate time lines, and separate resources were appropriate for the two PARs (and corresponding working groups). The new names for the working groups are checkpoint/restart (P1003.1m) and resource limits (P1003.1p).

The resource limits and checkpoint/restart sections of the POSIX.1a document will be removed from there and become the starting point for these new projects.

Other hot topics in POSIX.1a continue to cause long discussions, both in and out of the meeting room. How to deal with trailing slashes seems to be such a regular discussion

item that the vice chair has even suggested we add it as a fixed agenda item for every meeting!

The current standpoint on this, reflected in the last draft (draft 12), is to insist that a filename can have one or more trailing slashes only if it is a directory. The words currently talk about "as if a trailing /." were appended. This still needs some work, but it is unlikely that the next draft will be any more tolerant of trailing slashes on nondirectories than draft 12.

Another hot topic is the handling of group-ids when files are created. System V and BSD systems have totally different models here, and the original POSIX.1-1990 tolerated both. Attempts to settle on the BSD behavior have met with enough resistance that it now seems likely that both will continue to be tolerated.

Almost all this discussion is driven from the ballot process for POSIX.1a, currently trying to resolve issues on draft 12 and get a draft 13 out to ballot around the end of the year. It is a long, slow job, as anyone who has been through a ballot must realize, and most of us are new to it in POSIX.1a!

Sometimes changes are so deep into history that we cannot fathom the reason for them and have to start all over again. For example, some requirements on fflush were changed between 1988 and 1990, but no rationale was supplied at the time. What should be the effect of fflush on a stream opened for reading? Now we have to write a rationale for this action that occurred five years ago!

Report on POSIX.1e/POSIX.2c: Security

Larry Parker reports on the October 16-20, 1995, meeting in St. Petersburg, FL

Nine months ago the future looked relatively bright for the POSIX security working group. The resolution of all comments/objections from the most recent ballot had been completed in only three meetings, and a new ballot had been scheduled to occur with less than a year having passed since the previous ballot. Ballot resolution had proceeded well, and it appeared as if we might be approaching the point of being able to start the recirculation process by year's end. The only thing that needed to happen was for our technical editor to produce a formatted version of the draft by the end of January.

Well, it didn't happen. In fact, it didn't happen for six months!

Due to the massive delay caused by our now ex-tech editor, we have missed two scheduled ballot windows and currently can't obtain a ballot window until March 1996. If this doesn't change, we will have gone a full two years between ballots. This is an unacceptable delay in the ballot process

and has rightfully caused many to question whether the security working group is capable of producing a standard in any time frame. This concern has been heightened by the fact that the group has not met since January and no information on the group's status has been provided to the POSIX leadership. And, to make matters worse, one of the central members of the working group resigned due to a change in job responsibilities.

So what's happening now to correct this mess? Can the group pull itself together to complete the ballot process for the security standard? The following answers are strictly my own opinions. The work that has been accomplished to date on this draft standard through the work of many people over several years is far too important to allow it to be discarded due to the failure of one person. The core of the group has pulled together to put some life back into the effort. We have a new tech editor who produced a formatted draft within a week of receiving the materials from the previous editor. The group met in St. Petersburg in October and gave the draft a full review for consistency and updated much of the document rationale; the draft is in good shape, and we should still be able to start the recirculation process after the next (our fourth) full ballot. Essentially, we are ready for a ballot as soon as we can obtain a ballot window. Coincidentally, the POSIX organization is considering a change to the ballot scheduling process that may result in the draft being released for ballot as early as November of 1995! *[Editor's Note: Nick Stoughton reports that this has not happened as of January 5, 1996.]*

In short, the group has gone through a year of major turmoil but is beginning to return to the business of carrying the draft through the standardization process as quickly as possible.

Report on POSIX.1h: SRASS

Richard Scalzo <rscalzo@relay.nswc.navy.mil> reports on the October 16-20, 1995, meeting in St. Petersburg, FL

You say that your computer system fails without warning, that this happens too frequently, and that this state of affairs makes you nervous? You say that repairing it is too expensive? If you want to do something constructive about this problem, participate in the POSIX standards process. As the market for highly reliable computer systems grows, so will the need for portable applications that can manage faults for your system. The best part is that there is a lot of activity concerning standards for system services for fault management. You can get in on the ground floor (well, maybe the second floor) of this expanding activity if you hurry!

The POSIX.1h working group (Services for Reliable, Available, and Serviceable Systems, SRASS) is in the process of developing standard sets of APIs to support fault manage-

ment. The goal of the SRASS working group is to produce a coherent set of APIs that allows applications to perform fault management functions and to be portable.

Right now the SRASS working group is in the process of producing drafts of standard APIs for logging and notification, core dump control, and configuration management. These APIs, of course, are only part of the picture (more on that below).

The logging APIs are aimed at allowing an application to log application-specific and system events and for notifying applications when these events of interest occur. The functions are: write to the system log, open a connection to a log file, read from the opened log file, provide notification of events of interest, and find that part of the system log of interest to your application.

There is a single core dump control API to enable an application to specify a location for a process that terminates with a core dump file. The SRASS working group felt that your application should be able to find the core dump file in case you (unintentionally, of course) brought your system to its knees!

The proposed API set for configuration management is the most ambitious effort undertaken by SRASS to date. Its claim to fame is that it will allow an application access to underlying system configuration information that is available at boot time (which is normally invisible to an application). It will also allow an application access to those parts of the configuration space of a system that it may cause the system to generate. The primary purpose for the proposed interface is to support the recovery of a system from the effects of faults. In particular, the proposed set of APIs will allow an application to keep track of system configuration data that is important to recovery. It will allow an application to maintain a picture of the configuration of the system that is relevant to it. This is achieved by means of mount and unmount operations, linking and unlinking operations, operations to add nodes to the configuration description, and several functions to allow an application to access any part of the current description of the configuration picture.

The realtime contingent of the SRASS working group feels that there is a need for a set of APIs to help manage event detection. This is because realtime systems require more flexibility in interfacing with operating systems than do other types of application programs. Dr. Arun Chandra of IBM made a presentation on IBM Phoenix Event Management capabilities. These capabilities allow an application to access and manage system information on the state of system resources. It is hoped that these and the associated model will be made available for standardization shortly.

In addition, there is a lot of activity related to SRASS. There is a new working group for checkpoint/retry. This working group was formed after an original proposal was deemed to be incompletely specified. Because of the importance of checkpointing/restart to the high-assurance computing world, a new working group was formed. There is still a lot of work to do in this area, and your participation is invited.

Joint work with the realtime working group on event tracing is ongoing. So far, 35 requirements for tracing have been identified. There was lively discussion on the merits of tracing at the thread level and whether a trace on a process should span a fork. Several other requirements led to much debate before being resolved. A presentation on the use of trace facilities used by the ARPA HiPeR-D project was made by Eric Lager of the Naval Surface Warfare Center. He presented a description of requirements that the realtime community has for trace facilities. It was noted that in order to do trace in the stringent realtime environments of HiPeR-D, high-resolution clocks are required. It was decided that time stamps were an important part of being able to extract causality and order from a completed trace file, so the current requirement calls for time stamps to be made available. An initial proposal for trace APIs is expected to be ready in time for the January POSIX meeting. Attendance at these joint meetings was high and very active. There was lively participation by representatives of SUN, IBM, Tandem and Sequoia, as well as members of the realtime community. For more information on these activities, get in touch with Jim Shaffer at *jjs@austin.ibm.com* or Francois Riche at *rich@ibm.fr*.

To top things off, there was a presentation by Dr. Lonnie Welch of the New Jersey Institute of Technology concerning the need for the HiPeR-D Project to be able to access system statistics via an API. System statistics are needed to assess system performance, which must be analyzed before the trace facilities are used.

If you are interested in helping to produce standard APIs that support fault management (including serviceability and fault tolerance aspects of systems), get in touch with Helmut Roth at *hroth@relay.nswc.navy.mil* or Dr. Arun Chandra at *achandra@vnet.ibm.com*.

Report on POSIX.1m: Checkpoint/Restart

Steven J. Dovich <dovich@tiac.net> reports on the October 16-20, 1995, meeting in St. Petersburg, FL

The checkpoint/restart working group, otherwise known as POSIX.1m, began considering text extracted from previously balloted material from POSIX.1a. This was the first meeting of the group since the approval of the Project Authorization Requests (PARs) that split the content of the POSIX.1a draft. It seems that there was some expectation that the formation of the new working group meant that the

previous work was being discarded. The reality of the situation is that the new working group is using the text from the current POSIX.1a draft as its first draft. Whether preserving the investment in the languages of that draft is appropriate will doubtless become evident as this group brings a document forward through the balloting process.

The objections and comments from the last round of POSIX.1a balloting formed the basis for POSIX.1m group discussion. It seems strange to begin new working group activities with a ballot resolution. And I should note that none of these comments requires a response from POSIX.1m because they were submitted against a different draft document. Because we are all nice people, and because we recognize that these comments were offered in order to improve the language of the standard, we felt it important that each comment be considered and appropriate changes be made. Besides, if ignored, these objections and comments will probably be resubmitted as soon as this draft is sent out for balloting anyway (folks who join ballot groups can have rather long memories).

A portion of the comments was obvious enough that the group reached consensus on the appropriate changes in this meeting. Most of these dealt with ambiguity due to undefined terms, missing descriptions, or text that was acknowledged as unacceptable and for which an appropriate solution was supplied in the POSIX.1a ballot. There remains a list of issues that have been deferred, either because of the complexity of the proposed solution or because of the subtlety of related issues already documented in the published standards.

A sufficient number of items was agreed to in this meeting to provide plenty of work for the technical editor and other group members, during the next few months. Barring any issues other than those already identified from the POSIX.1a ballot, this working group should be able to prepare a draft ready for balloting by the end of next year.

Report on X/Open Distributed Systems Management

Martin Kirk <m.kirk@xopen.co.uk> reports on X/Open Distributed Systems Management

The X/Open Distributed Systems Management Program commenced in 1990 and aims to progressively define an environment for the development of distributed management applications for heterogeneous systems. The program has produced a variety of deliverables, including a Systems Management Reference Model, the XMP Management Protocols API, a first volume of Common Management Facilities for an OMG CORBA (Object Management Group's Common Object Request Broker Architecture) environment, and specifications for Performance Measurement and Backup and Restore Services.

Current and future activities include the definition of further Common Management Facilities, Event Management Services, and Distributed Software and Print Administration.

The work of the X/Open Systems Management working group is complementary to other activities such as IEEE POSIX P1387, the Network Management Forum (NMF), the Object Management Group (OMG), and the Performance Management Working Group (PMWG).

X/Open's role in distributed systems management is to promote the convergence of management protocols and object definitions, the establishment of a consistent management environment on open systems, and the integration of open systems as both client and management application platforms in networks of heterogeneous computing environments. In this role, X/Open aims to serve as a facilitator, adopter, integrator, and/or innovator to promote agreement and rapid implementation and deployment of distributed management services.

The X/Open Distributed Systems Management program is concerned with the distributed management of distributed, heterogeneous systems, covering:

- Management of Stand-alone Systems

- Management of Distributed Systems
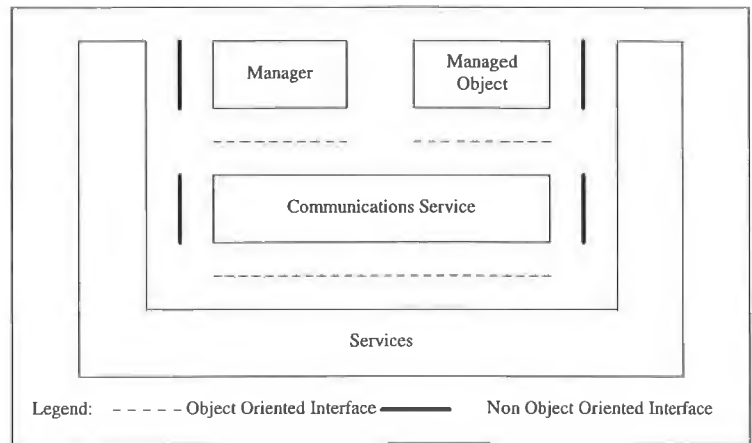
- Application Management

- Network Management

It aims to produce the specifications necessary to facilitate the development of systems management software for a distributed, heterogeneous environment.

The program has the following broad goals:

- Portability of Management Software

- Portability of Administrators

- Interoperability of Management Systems

- Integration between Systems and Network Management

In addition to the user requirements developed as part of the X/Open's requirements gathering process, the working group works with the X/Open Distributed Systems Management Requirements Topic Group to identify and refine the detailed requirements that shape the continuing technical program.

X/Open has defined a framework for its current specification development work in this area in the Systems Management Reference Model. The following diagram taken from that model illustrates the overall concepts involved:

Legend: – – – – – Object Oriented Interface ———— Non Object Oriented Interface

The diagram illustrates the relationship between managers (who implement the management tasks performed by administrators) and managed objects (which represent the resources being managed).

Communications between managers and managed objects is provided by a Communications Service, which also provides access to other services necessary to implement distributed management systems.

Services can be divided into the following broad classes:

- general services, which are characterized as being of use to a wide range of different problem areas

- management services, which are common facilities that have been specialized for distributed management (Areas of specialization might include policies for more centralized control of security, policies for configuring and distributing applications, and the ability to control the location of objects.)

- application services, which are services specific to some particular functional area within the overall management problem space (Although these services are not of general use to a wide range of management applications, they provide common services to implementations addressing that particular area. An example might be a catalog service provided for the use of multiple backup and restore applications.)

The interface to the Communications Service implements an object-oriented paradigm. The interfaces provided by other services may also be expressed in the same way; however, some service interfaces will be defined as non-object-oriented, functional interfaces. The reason for this approach is wholly pragmatic; object-oriented frameworks are not universally deployed, and in order to deliver specifications in a timely manner, it is not possible to predicate them on the existence of object-oriented framework technology. The reference model is deliberately described using abstract terminology, independent of any specific implementation

technology. At present, a variety of technologies is in common use:

* SNMP (Simple Network Management Protocol) and CMIP (Common Management Information Protocol) Management Protocols for Network Management

* RPC (Remote Procedure Call) for Systems Management

* OMG CORBA for emerging Systems and Network Management frameworks

The X/Open Distributed Systems Management Program will incorporate the above technologies. X/Open has defined an API specification (XMP) that provides consistent access to the SNMP and CMIP management protocols and an accompanying specification (XMPP) that references the protocol specifications supported by XMP.

Current development of "application-level" specifications is being performed using RPC technology as the underlying mechanism. As noted above, these developments indicate the need to respond pragmatically to user requirements and in as timely a manner as possible.

Work is under way to develop interfaces to management services using the OMG CORBA technology. Where applicable, a migration path will be provided for RPC-based specifications to a CORBA environment.

Work is also under way to enable effective interworking between network management frameworks based on SNMP and CMIP, and the OMG CORBA technology that is expected to form the basis of future systems management frameworks.

The systems management working group collaborates with several other related groups, including NMF, OMG, and PMWG. This integration role is an important part of the X/Open strategy, and further collaborative relationships with other groups are expected in the future.

X/Open published several classes of document:

Snapshots: These provide a mechanism for X/Open to disseminate information on its current direction and thinking. The intention is to stimulate industry debate and prototyping and solicit feedback. A snapshot represents the interim results of an X/Open technical activity. A snapshot does not represent any commitment by X/Open members to develop any specific products.

Guides: These provide information that X/Open believes is useful in the evaluation, procurement, development, or management of open systems, particularly those that

are X/Open-compliant. They are advisory, nonnormative documents.

Preliminary Specifications: These specifications, which often address an emerging area of technology and consequently are not yet supported by multiple sources of stable conformant implementations, are released in a controlled manner for the purpose of validation through implementation of products. A preliminary specification is not a draft specification. In fact, it is as stable as X/Open can make it and on publication has gone through the same rigorous X/Open development and review procedures as a CAE specification. Preliminary specifications are analogous to the trial-use standards issued by formal standards organizations, and product development teams are encouraged to develop products on the basis of them. It is expected that preliminary specifications will normally progress to become CAE specifications once suitable implementation experience has been gained.

CAE Specifications: CAE (Common Applications Environment) specifications are the stable specifications that form the basis for X/Open-branded products. These specifications are intended to be used widely within the industry for product development and procurement purposes.

The following publications developed in the Distributed Systems Management Program are currently available:

* S110. Systems Management: Problem Statement. 8/91

* S190. Systems Management: Identification of Management Services. 5/92

* G211. ISO/CCITT and Internet Management: Co-existence and Interworking Guide. 12/92

* G207. Systems Management: Reference Model. 9/93

* G302. Systems Management: Managed Object Guide. 9/93

* G141. Systems Management: Guide to the Universal Measurement Architecture. 12/94

* P426. Systems Management: UMA Measurement Layer Interface. 12/94

* P434. Systems Management: UMA Data Capture Interface. 12/94

* P435. Systems Management: UMA Data Pool Definitions. 12/94

* P424. Systems Management: Backup Services API. 7/95

* P421 Systems Management: Common Management Services, Volume 1. 7/95

- P521. File System and Scheduling Utilities. 10/95

- C206. Systems Management: Management Protocol Profiles (XMPP). 11/93

- C306. Systems Management: Management Protocol API (XMP). 3/94

- C502. Systems Management: GDMO to XOM Translation Algorithm. 10/95

More detailed information is available on the X/Open Web server, URL: *http://www.xopen.org*.

The following projects are currently under development within the systems management working group:

- *Common Management Facilities, Volume 2.*
  Preliminary specification: 2Q96.
  This project will add to the OMG IDL-based (Interface Definition Language) services defined in Volume 1.

- *Event Management Service.*
  Preliminary specification: 3Q96.
  This project is intended to define an event management service capable of receiving events from a variety of sources and providing mechanisms by which an application can register to receive events in which it is interested.

- *Distributed Software Administration Interoperability.*
  Preliminary specification: 4Q95.
  This project will develop interoperability interfaces that will be complementary to the POSIX P1387.2 software administration standard. The POSIX standard concentrates on issues of portability, and the X/Open specification will provide an interoperability definition that will allow the development of distributed, heterogeneous software administration.

- *Distributed Print Administration Interoperability.*
  Preliminary Specification: 2Q96.
  This project will develop interoperability interfaces that will be complementary to the POSIX P1387.4 print administration standard. The POSIX standard concentrates on issues of portability, and the X/Open specification will provide an interoperability definition that will allow the development of distributed, heterogeneous print administration.

- *Inter-Domain Management: Specification Translation*
  Preliminary Specification: 4Q95
  Inter-Domain Management: Interaction Translation
  Preliminary Specification: 2Q96
  This project, undertaken in collaboration with the Network Management Forum, will establish guidelines for translating managed object definitions between ISO

GDMO and SNMP, and OMG IDL. This will enable the simpler interworking of management systems based on ISO and SNMP and OMG technology. It is expected to be particularly important in enabling better integration between systems and network management.

- *UMA Data Capture Interface.*
  CAE specification: 3Q96.
  UMA Measurement Layer Interface.
  CAE specification: 3Q96.
  UMA Data Pool Definition.
  CAE specification 3Q96
  These deliverables represent the completion of the X/Open process for the existing preliminary specifications. These specifications were developed by the Performance Management Working Group. They define interfaces and metrics for performance measurement.

Until relatively recently, the Distributed Systems Management working group consisted primarily of representatives of the major system vendors who are the X/Open shareholders, together with representatives of the X/Open user and ISV councils. In 1993, X/Open created a new form of membership that allows participation directly in individual working groups, and this has resulted in a significant number of systems management ISVs joining the group. For further information on either membership or the work of the Distributed Systems Management working group, please contact the author, *<m.kirk@xopen.co.uk>*.

Open Systems, POSIX, and Windows NT – Another Point of View

by Heinz Lycklama
<heinz@osta.com>

"It's Official: Windows NT Is Open" – Editorial by Michael Goulde in the November 1995 issue of *Open Computing*

"Feds declare NT 'open system'; UNIX takes a hit" – *ComputerWorld* news headline, July 31, 1995

"NT is a FIPS-2 certified system, and as such is a 'POSIX-compliant' operating system" – stated as fact in "redacted decision" by judge from GSBCA

What's going on here? What do these statements from recent trade publications and the judge's "redacted decision" have to do with the facts? Are any of them true? How did the GSBCA judge come to this conclusion in the Coast Guard Standard Workstation III award? The flurry of activity following the US Government bid protest judgment

handed down in June 1995 has been remarkable. Are thirty lawyers better qualified to define an "open system," let alone a "POSIX-compliant operating system," than the technical experts who produced the POSIX standards and the POSIX.1 Testing Policy?

What we have here is the trade press badly misrepresenting the decision that was handed down by the US Government, analysts repeating what the trade press is reporting without doing any real analysis, the judge in this case making statements contrary to the spirit and law of the NIST POSIX.1 Testing Policy, and at least one "POSIX expert" agreeing with the judge's statement, even though it is contrary to the NIST POSIX.1 Testing Policy.

The press has done a great job of clouding the issue of "open systems." First it was *ComputerWorld* with its article stating that the government declares that Windows NT is "open." The November 1995 issue of *Open Computing* has an editorial written by Michael Goulde of the Patricia Seybold Group with the title: "It's Official: Windows NT Is Open." Michael states that the "GSA Board of Contract Appeals declared that Microsoft Windows NT is an open system." They said no such thing! Where do these analysts/reporters get this from? What's a user to believe?

In his article "Open Systems, POSIX, and NT," published in the December, 1995 issue of *;login:*, Stephe Walli provides much of the background of this protest case involving the award of the US Coast Guard Standard Workstation III (CGSW) RFP to Unisys. He also provides a summary of the "findings of fact" with some discussion from the judge's redacted decision. I won't bother to repeat the "findings of fact," but I do disagree with his conclusions. This article explains my views on why the decision handed down in this protest bid is incorrect and sets a bad precedent for those who promote the use of open systems – suppliers and users alike.

I was directly involved in the bid protest trial as an expert witness on POSIX-related issues for the protesters (proud to represent the side of "open systems," I might add). As the founder of the original */usr/group* Standards Committee, which spawned the IEEE POSIX standards efforts, I care a great deal about how POSIX and open systems are viewed and used in the industry.

Someone familiar with this protest said, "Now that Windows NT has won a large bid by following the rules that the UNIX community created, UNIX people are crying foul – That's not right!" My contention is that Windows NT "won" this bid by the not-so-subtle use of a "bait-and-switch" policy, and not by rules that the UNIX community created. Let's look a little deeper into the issues of POSIX compliance and "open systems" surrounding this protest.

Is Windows NT POSIX-compliant?

For that we need to go to the NIST POSIX Testing Policy. This policy recognizes that the POSIX.1 APIs can be hosted by a number of different configurations, one being a "cooperating-hosted system." In the definition of terms, it states that a "cooperating-hosted system" is "a single computer system that provides the functionality of both the development system and the host implementation with a *single* operating system, and provides the FIPS 151-2 conforming implementation with *another* operating system" (emphasis added). This definition was introduced to accommodate the testing of the Windows NT POSIX Subsystem. No problem here – the intent is clear when you look at the three other configurations that had been dealt with by the NIST POSIX Testing Policy heretofore (native implementation, hosted implementation, and cooperating system). Windows NT supports multiple operating system environments, e.g., Win32, OS/2, and POSIX, and thus a new test configuration definition was required.

In the Certificate of Validation issued by NIST, the implementation tested is the "Microsoft Windows NT POSIX Subsystem, Version 3.5." It should also be noted that there are some major deficiencies listed on the Certificate of Validation, including the following:

- general terminal interface devices

- mountable file systems

- modem control

- appropriate privileges

These deficiencies carry no legal binding, but they do indicate that the POSIX Subsystem of Windows NT barely squeaked through the tests.

The Windows NT POSIX Subsystem is the validated product, the "another operating system" in the definition of cooperating-hosted introduced in the NIST POSIX Testing Policy. The "single operating system" in this case is the Windows NT Win32 Subsystem – that is the development system that was used to compile the POSIX test suites. The implementation under test, i.e., the validated FIPS 151-2 product, as identified in the NIST POSIX Testing Policy and in the Certificate of Validation is the "Windows NT POSIX Subsystem." The NIST POSIX Testing Policy says that "The product identified represents the operating system tested." This is correctly identified in the Certificate of Validation as the "Windows NT POSIX Subsystem."

Because the "Windows NT POSIX Subsystem" is certified to be POSIX compliant, does this mean that Windows NT is POSIX compliant? No, the definitions in the NIST POSIX

Testing Policy are very clear that this is not what is meant. NIST never intended it this way, and NIST personnel have testified to that. Consider this analogy with the cooperating system in which the development system (which is used to compile the POSIX test suites) and the target system (which is used to run the POSIX test suites) are two separate computers. If the target system is certified to be POSIX compliant, does that make the development system POSIX compliant? I don't think anyone would argue that, but that's exactly what is being claimed for the Windows NT system. The claim is that because the POSIX Subsystem is POSIX compliant, Windows NT is POSIX compliant.

Did the Windows NT-based solution proposed meet the CGSW III RFP?

One of the major requirements of the CGSW III RFP is that certain applications (email and RDBMS) run under the POSIX operating system. We interpret this to mean that these applications must run under control of the POSIX compliant operating system. For the Windows NT platform proposed, that means the POSIX Subsystem, which is the operating system environment that provides the POSIX.1 services. The email and RDBMS products proposed run in the Win32 Subsystem. So how does this meet the requirements of the RFP?

One of the other major objectives of the RFP was to provide a platform for portable applications using the NIST Application Portability Profile (APP) as a framework. Certain standards were selected from this APP for the CGSW III RFP. These include:

- GOSIP (FIPS 146-1)

- SQL (FIPS 127-1)

- XVT

- C (FIPS 160)

- Ada (FIPS 119)

- Pascal (FIPS 109)

- POSIX.1 (FIPS 151-2)

The intent of the NIST APP (and of the POSIX Open System Environment (OSE) as defined in the POSIX.0 Guide for Open Systems Environment, upon which the NIST APP is based) is that the APIs defined by the standards be part of an integrated environment so that a portable application can use any and all of the APIs that are part of the APP. The Windows NT-based solution provides the GOSIP, SQL, XVT, C, Ada, and Pascal standards in the Win32 Subsystem and only C and POSIX.1 in the POSIX Subsystem. This makes it impossible to write a portable program that uses all these APIs in an integrated manner so that the application can be ported to

another POSIX.1 compliant platform. So the solution proposed defeats the intent of the NIST APP, the government's own proposed framework for developing portable applications.

Clearly, the proposed Windows NT-based solution does not meet the letter, intent, or spirit of the CGSW III RFP. How did this happen? If the CG wanted Windows NT, they should not have written "POSIX operating system" into the requirements, or determined a need for portable applications for that matter. They should have stated up front that a proprietary solution such as Windows NT was acceptable. Other bidders spent millions of dollars to put together bids that complied with the POSIX and portability requirements.

The government, NIST specifically, spent millions of US citizens' tax dollars to define procurement procedures that would meet the needs of various government agencies. Part of the effort was to define an Application Portability Profile that would provide a framework for writing portable applications, and give the agencies the choice of selecting from multiple suppliers, knowing that their current applications would still run on any new platform that they might acquire in the future. This is called investment protection.

Investment protection was in fact a major objective for the CG because they wanted to move from a proprietary CTOS environment to an "open environment" that would give them choice of suppliers and solutions in the future. The CG has moved from one proprietary platform to another with the Windows NT solution. The CG also wanted to be able to import applications written by other government agencies for their "open platforms." The Windows NT-based solution also defeats this purpose because portable POSIX compliant applications cannot be ported to the Windows NT platform.

We even heard the argument that "The language 'run under' was used by the Coast Guard to prevent bidders from proposing solutions of these applications that were run under emulation." Give me a break! The English language is not that imprecise that one would believe that the word "POSIX" was introduced so that we should interpret "POSIX Operating System" as meaning that this disallows emulation. There is no mention of emulation in the RFP. This really stretches credibility! In fact, even the Win32 Subsystem in Windows NT supports Windows applications running in 16-bit mode using emulation. This interpretation of the RFP language would even disallow Windows NT as a solution!

Microsoft has a desire to capture as much of the government market for computing platforms and applications as possible. (They have that right, but they need to play by the same rules as other suppliers do.) Windows NT was designed with the government market in mind. As stated in chapter 1 of the book *Inside Windows NT* by Helen Custer of Microsoft,

"To meet the government's POSIX procurement requirements, NT would be designed to provide an *optional* POSIX application execution environment" (emphasis added). This is exactly what Microsoft has done – added an optional POSIX Subsystem for no purpose other than passing a POSIX.1 test suite. No commercial Microsoft or third party products that use the POSIX Subsystem have been introduced. It was never intended to be useful. In fact, one can remove the POSIX Subsystem, and all commercial applications will run just fine.

Let's call a spade a spade. Bottom line, this is a "bait-and-switch" policy: the POSIX Subsystem is the bait, and the Win32 Subsystem is the switch. "Yes, we have POSIX, but please use our Win32 Subsystem, i.e., Windows Open System Architecture, instead – it's the only one that really works." Is this a marketing sham or what? It's like writing a contract to have a house built with 110 volt sockets. Your contractor builds the house with 220 volt sockets, but with only two 110 volt sockets. The test is whether your toaster will work on the 110 volt socket. Yes, but if you plug in two appliances on the two 110 volt sockets, a fuse is blown. Oh by the way, you can plug in as many 220 volt appliances as you want, but you can only buy those appliances from a factory in Redmond!

What makes a system open?

That depends on who writes the definition. Portability, interoperability, and user portability are three agreed-upon key requirements. How well do products on the market meet these requirements? This has become a very subjective discussion. The POSIX Open System Environment has four major goals:

- application portability

- application interoperability

- data portability

- user portability

with the resulting benefits of:

- integration of components from multiple vendors

- efficient development and implementation

- efficient porting of applications

The NIST APP, which has a strong resemblance to the POSIX OSE, was developed for the government market to make large government procurements more cost-effective and efficient and to promote portability and interoperability between IT solutions adopted by various government agen-

cies. The key here is that suppliers and users must agree on an application framework to meet the stated goals and achieve the benefits listed above.

Computing systems and applications that meet the above-stated goals meet the needs of the government agencies. Application portability works only if the application uses an integrated set of APIs that fits within a well-defined applications framework such as the NIST APP or POSIX OSE. The X/Open application profiles also match the NIST APP and POSIX OSE very well. Most UNIX systems, and even proprietary operating systems with integrated "open systems environments," delivered today provide a consistent set of "open systems" APIs agreed to by players in the open systems industry. These systems provide open platforms suitable for applications portability and interoperability.

Given these application profiles/frameworks, openly defined by all participants in the process, any system vendor can build computing platforms to meet the requirements, and any ISV can build applications that fit into the framework. The user has a choice of system providers and a choice of applications providers. The framework is open and not controlled by any one vendor. This model fits the government's standards-based procurement needs and does not lock the government into any one vendor. This is openness in the purest sense of the word. The specifications for all important "open system" component interfaces such as POSIX, X windows, TCP/IP, CORBA, and now the World Wide Web were determined by cooperation among industry suppliers.

Is Windows NT "Open"?

By whose definition? Does it support application portability? Only if you move an application from one Windows NT platform to another. Porting an application from Windows NT to a UNIX system or vice versa is not easy because the set of APIs used on one system is not necessarily supported on the other system. UNIX systems provide an integrated set of APIs that matches the requirements and framework of the NIST APP. Windows NT provides a different set of APIs that does not meet the framework requirements of the NIST APP, but rather fits within the Microsoft-defined WOSA. The POSIX.1 APIs provided by the Windows NT POSIX Subsystem do *not* fit into the WOSA framework (by design).

With Windows NT, we have a model where the application architecture or framework (WOSA) and the APIs are controlled by one vendor – Microsoft. The user can buy computing platforms and applications from any supplier who provides Windows NT and applications that fit the WOSA architecture, all owned by one vendor. Open systems is an attempt to break this control by one vendor.

By adding the POSIX Subsystem as an appendage to Windows NT and then declaring that Windows NT itself is POSIX compliant, Microsoft has corrupted the concept of "open." Microsoft advertises Windows NT as a POSIX-compliant operating system, thereby subverting the meaning of POSIX compliance. Other operating systems suppliers such as IBM, HP, DEC, and Tandem have added POSIX.1-compliant interfaces to their proprietary systems in an integrated manner, but these POSIX.1-compliant environments were meant to be, and are, used by their customers to build portable and interoperable applications. Microsoft has no such intent in providing the POSIX Subsystem appendage to Windows NT.

Users are free to buy proprietary solutions controlled by one supplier. But if this is what the user intends to procure, then the RFP should state clearly that all solutions will be considered, open or proprietary. Don't use POSIX compliance as a ruse. Don't even use the term "POSIX compliant" in the RFP if it carries no meaning.

Possible Responses by the Open Systems Community

The CGSW III RFP was awarded improperly to a supplier that responded with a noncompliant operating system, Windows NT. It is a mistake to let this award stand because of the precedence it sets. Here are some options that suppliers interested in open systems have:

- Mount a concerted effort to overthrow this award. Letting it stand confuses the meanings attached to "POSIX compliant" and "open" in the user's mind. From a technical point of view, the US Government's ruling doesn't have a leg to stand on. The "finding of fact" quoted above is in fact false, according to the NIST POSIX Testing Policy.

- Mount an open systems marketing effort to shed light on what's really happening in order to educate confused users. This will encourage users to write RFPs that result in the procurement of open systems solutions. RFPs must be written with a lot more precision than they have in the past. Open systems give users choice.

- Work with users to strengthen the demand for open systems solutions. X/Open, OSF, and UniForum are in a position where they can help influence the writing of RFPs that result in the procurement of open systems solutions. RFPs must be written with much more precision to avoid the problems encountered with the CG III RFP.

- Work with NIST to strengthen the RFP requirements writing procedures to assure that the government acquires open systems solutions that meet the NIST APP specifications. The government has spent millions of dollars to develop the NIST APP and the test suites that are used to

measure conformance. Let's not let this effort go to waste.

- Strengthen industry cooperative efforts to avoid unnecessary fragmentation and to counter the inroads being made by Windows NT. Industry players have taken a number of steps to strengthen the role of open systems technologies in the past year. We need more open systems technologies such as X windows, TCP/IP, NFS, World Wide Web, CORBA, and Java to give users a choice in buying open systems solutions from more than one vendor.

Postscript

"Michael Goulde recants statement" – *Open Computing*, December 1995 issue, page 10

"Open Computing magazine closes its doors" – *Unigram X*, Issue 567, December 4-8, 1995

"Stephe Walli does penance – builds the real McCoy" – Details at UniForum'96 in San Francisco.

Sun Microsystems pours hot Java on Microsoft and writes script for the new game.

Microsoft: "Let's hope Anne Bingaman [DOJ] doesn't read about this."

30 lawyers agree to redefine "open system" by spelling the second word as s-e-a-s-o-n.

BOOK REVIEWS

The Bookworm

by Peter H. Salus
<peter@uunet.uu.net>

It's really nice to know that folks read this column: I got flak from several different publishers about my remarks on dust jackets (they all seemed to think that I was some sort of nonagenarian); I also got mail from readers (a.k.a. live human beings) who thought I was right. And there were several complaints about those plastic laminated covers that so many publishers now use. But it was gratifying to get the email.

Big Nets and Small Nets

When Steve Crocker wrote RFC 1 in early 1969, the NWG had allocated 5 bits for addressing; by the time the first IMP was installed in Kleinrock's lab at UCLA (September 2, 1969), it was 6-bit addressing; two years later it was 8 bits; by 1974, we had the beginnings of TCP/IP and 32 bits. It was clear by 1991 that DNS and 32 bits were going to be inadequate. So the IAB set up a task force headed by Scott Bradner and Allison Mankin. Effectively, the work is over. We will be going to 128-bit addressing. Bradner and Mankin have put together an anthology that attempts to explain the process; the only thing they've left out is just how painful it's going to be.

Some of you may recall January 1, 1983, and its aftermath. This will be less painful, thanks to the IAB's task forces. And the transition will get us ready for 256- or 512- bit address space. Don't kid yourselves: 128 bits won't be enough.

This is a fine volume and a must read for anyone involved with site management where there's an Internet connection.

At the other end of the scale, back in the early 1970s, Bob Metcalfe and Dave Boggs invented Ethernet. Today there are about 50,000,000 computers that are on Ethernets. Spurgeon has written a small, reasonably-priced volume that will end up being indispensable for any LAN administrator. I have found IEEE 802.3 unreadable. Spurgeon makes the configuration rules intelligible. And he's got Metcalfe's 1976 drawing on the cover!

Shells

A while back I complained that there was no book on bash or on zsh. Well, now there's one on bash. I'm still not enamored of it, but it is a useful, POSIX-compliant shell. (I find that I use sh at least once or twice a day, I admit.) But bash is the default shell for Linux, so it can't be all bad. Newham and Rosenblatt have done a very fine job here. (Rosenblatt is also the author of O'Reilly's ksh book; there are most likely nutshells all over his floor.)

More Linux

I mentioned both the InfoMagic and the PTF disks in my last column. So I was given the Red Hat Commercial Linux set at FedUNIX in Washington, DC. It's quite a package. You get a 4 CD set of Red Hat Linux 2.0, a Live File System: "Run Linux from Your CD drive," the TSX-11 Linux and GNU Archives, and the Sunsite Linux Archives. You also get a 120-page user's guide. All for $49.95. The quick install works; the chapter of typical questions actually answers those questions. And it's only 120 pages, so you can easily RTFM!

Books reviewed in this column:

Scott Bradner and Allison Mankin, eds., **IPng: Internet Protocol Next Generation.** Reading, MA: Addison-Wesley, 1995. ISBN 0-201-63395-7. Pp. 336. $33.30.

Charles Spurgeon, **Ethernet Configuration Guidelines.** San Jose, CA: Peer-to-Peer Communications, 1996. ISBN 1-573398-012-9. Pp. 178. $19.95.

Cameron Newham and Bill Rosenblatt, **Learning the bash Shell.** Sebastopol, CA: O'Reilly & Associates, 1995. ISBN 1-56592-147-X. Pp. 310. $27.95.

Red Hat Commercial Linux. Westport, CT: ACC Bookstores, 1995. 4 CDs + 120pp. book. $49.95.

Arthur van Hoff, Sami Shaio, and Orca Starbuck, **Hooked on Java.** Reading, MA: Addison-Wesley, 1995. ISBN 0-201-48837-X. Pp. 208 + CD-ROM. $29.95.

Paul Graham, **ANSI Common Lisp.** Englewood Cliffs, NJ: Prentice Hall, 1996. ISBN 0-13-370875. Pp. 430.

Steve Summit, **C Programming FAQs.** Reading, MA: Addison-Wesley, 1996. ISBN 0-201-84519-9.

Gregory Satir and Doug Brown, **C++: The Core Language.** Sebastopol, CA: O'Reilly & Associates, 1995. ISBN 1-56592-116-X. Pp. 228. $19.95.

Ed Krol and Pamela Ferguson, **The Whole Internet for Windows 95.** Sebastopol, CA: O'Reilly & Associates, 1995. ISBN 1-56592-155-0. Pp. 650. $24.95.

Paul Gilster, **The New Internet Navigator.** New York: John Wiley, 1995. ISBN 0-471-12694-2. Pp. 735. $24.95.

Allan Leinwand and Karen Fang Conroy, **Network Management.** Reading, MA: Addison-Wesley, 1996. ISBN 0-201-60999-1. Pp. 338. $39.76.

Bruce Schneier, **Applied Cryptography.** New York: John Wiley, 1996. ISBN 0-471-11709-9. Pp. 758. $49.95.

This having been said, if you're not a tinkerer or a sys admin, Linux may be "fun," but I fear that it isn't ready for commoners to use. No support; and there are occasional questions that aren't in the FAQs. In the late 1970s, USENIX was the place that gave out the information that frustrated UNIX-users needed. Cygnus offers Linux support, but if you "subscribe" the OS doesn't look that cheap after all. BTW, I remarked in December that InfoMagic's 4-CD set could crash in an ugly fashion; I didn't push Red Hat that far. If you once built a Heathkit or a Dynakit hi-fi component, Linux is the thing for you.

Pouring Out Java

Java was originally called Oak by James Gosling. When it was noted that there already was a programming language named Oak, the name was changed during a trip to the local coffee shop. This makes one speculate on the names of successor languages: Kenya? Sumatra? Mocha? A stripped-down version that runs on the 386i called De-caf? The mind boggles.

Hooked on Java is the right stuff. Written by three members of the Sun Java team (Arthur van Hoff, Sami Shaio and Orca Starbuck), the 200 pages of *Hooked on Java* are truly first rate. I zipped through the book with frequent reference to the CD: all the applets in the text are on the disk, which also contains the Java developers kit and Java source.

The authors give all sorts of instructions on how to do animation, enhanced graphics, special typography, etc., but I didn't try anywhere near everything. However, what I did try worked, and attempting to alter an applet so that it would do what I wanted worked well, too. (Incidentally, *Hooked on Java* defines an applet as "small programs written in the Java programming language" (p. 2). That's a pretty good definition.

Chapters 5 and 6 of this book were, to me, the important ones: they discuss the language in reasonable depth and provide the instructions for building your own applets. This book and CD are really good.

Languages

I guess Java should be here, too. Well, right now it's special. In some ways, I guess that Lisp is special, too. It's the oldest computer language after Fortran that's still in use. I was a Lisper, using McCarthy's 1.5 in the mid-1960s; I've followed Touretsky's and Foderaro's Lisps. And here's a really nice presentation of ANSI Common Lisp by Paul Graham. It reads well. Graham's only shortcoming is not having a real bibliography (as opposed to end-of-chapter notes) and in never mentioning Franz Lisp (though there's a quotation from John Foderaro) or Richard Stallman (though there's mention of GNU emacs).

Steve Summit has put together a useful volume of *C Programming FAQs*. Once you've gone through Kernighan and Ritchie enough times, you don't have a lot of questions left. But the language impaired just beginning C will find this very handy. (Interestingly, it's nearly double the length of K&R.)

C++: The Core Language is intended to bring C programmers up to speed quickly where C++ is concerned. I think that Gregory Satir and Doug Brown have produced a useful book for those in transition, but programmers may want to transition to Java.

While I was in the Netherlands last November, I acquired *Programming Language Essentials* by Bal and Grune on the recommendation of Jaap Akkerhuis. This is a first rate small book on a variety of programming languages, both standard ones (like C or Prolog) and nonstandard ones (like PIC and SETL). I'm told that it's not available in the US. On the other side of the Atlantic, it's produced by Addison-Wesley. The ISBN is 0-201-63179-2. Another example of good marketing from a major publisher! I don't understand this at all.

Vahalia's *UNIX Internals* (Prentice Hall; ISBN 0-13-101908-2) is a fine book. I wrote the Foreword, and so will abstain from any other comment.

Revisitations

There are four volumes that are in their second (or greater) editions that are worth mentioning this month. Gilster's *The New Internet Navigator* and Ed Krol's *The Whole Internet for Windows 95* are both what they were before: first rate guides by truly knowledgeable authors. Ostensibly, the Krol is a new book, complete with a co-author and Windows 95 information. It's really a customized version of Ed's old standby.

Leinwand and Conroy's *Network Management* has added stuff on OSF's DME and SNMPv2, making the new edition more useful than its predecessor.

The new edition of Schneier's *Applied Cryptography* has moved us into the world of massive from merely large. If you have tender wrists, don't try to hold this while reading it. But you will want to read it; it's certainly the very best book on cryptography you can get.

There is an inordinate number of books called *The Object-Oriented Guide to Networking Client-Server Systems with Windows95 for Grandparents*: They're all trash.

Building Internet Firewalls

by D. Brent Chapman and Elizabeth D. Zwicky, O'Reilly & Associates, 1995, ISBN 1-56592-124-0, 544 pp., $29.95

Reviewed by Shawn Instenes
<shawni@istus.rain.com>

I've been building firewalls for a few years now. When I started, there were few commercial products and no public information on how to do it yourself; you had to know how to write least-privilege network code and how to fit it to your own network. Not only did the firewall do-it-yourselfers have to know how to put things together; they had to build their own tools. Things have changed.

For any firewall book, there's going to be a certain amount of comparison to Bellovin and Cheswick's *Firewalls and Internet Security*, so I'll get it out of the way. Bellovin and Cheswick's book brought a lot of firewall background information together; much here was presented in various tutorials and papers at USENIX conferences. You'll find a true-life story of an attack on their site. You'll find a discussion of the legal issues of monitoring intruders. However, the book was limited: it discussed the configuration and experiences of AT&T. But what about alternate configurations?

Where Bellovin and Cheswick's book has more background and theory, Chapman and Zwicky's *Building Internet Firewalls* contains detail. The first few chapters build background for what is to come, but the meat is in chapter 8. If you know what you want to allow through your firewall (and there is a lot of advice on this topic, about the most common protocols), you can look up what sort of things to do about it: packet filter details, proxy characteristics, what you might have to do to clients to work with these, and a summary of recommendations. There's also two sample firewall configurations in chapter 9.

Chapter 10 is the best single text I've ever read on authentication issues. Here we have hijacking and packet sniffing dangers discussed and what thwarts those attacks (only end-to-end encryption will completely eliminate the threat of connection hijacking). One-time passwords are explained in detail, as are time-based and challenge-response tokens.

One gem is chapter 11, which gives valuable advice on producing a security policy. A firewall is simply a mechanism for enforcing a certain security policy at a network trust boundary, so if you don't have a security policy, you really don't know what you're protecting, or why.

Many system administrators who think about putting up a firewall fail to provide for maintenance. Logs must be watched, new attacks are being carried out, and new tools become available. There's information about how to keep your firewall and yourself up to date in Chapter 12.

Finally, chapter 13, "How to Prepare for (and How Not To Panic) When an Incident Occurs." Should you disconnect? Should you watch? Who else needs to know when an incident is in progress? There's lots of advice here.

There's more than enough detail to build a firewall most anywhere. What's left out are items that really aren't standard: detail on setting up virtual private networks (there's a discussion of what they are and a figure) and dealing with "transparent" firewalls that some commercial vendors sell now, for instance. These are really nits, though. If you're concerned about these things then you're likely building firewalls for a living and are subscribed to Brent's firewalls mailing list already.

I said it last issue and I'll repeat it: this book belongs on the shelf of anyone who builds network firewalls.

Information about *Building Internet Firewalls* and the firewalls mailing list can be found at Great Circle Associates: *http://www.greatcircle.com*. The firewalls FAQ can be found at: *http://www.iwi.com/pubs/faq.html*.

To Engineer is Human: The Role of Failure in Successful Design

by Henry Petroski, St. Martin's Press, 1985, ISBN 0-312-80680-9, $18.95

Reviewed by George W. Leach
<gwll@gte.com>

I read this book many years ago and recently picked up a copy in the bargain bin at a local bookstore. Although it is not a software book, it does shed light on similar problems in an older discipline, structural engineering. The intent of this book is to explain in layman's terms why structural failures occur in the first place, even in the twentieth century, and how the structural engineering community learns from these experiences and progresses.

The author explores issues surrounding novel designs that failed and how the structural engineering community analyzed the problems. The engineering profession is able to progress by disseminating information to ensure that such failures do not occur again.

Among the specific designs described in the book are the Kansas City Hyatt Regency Hotel skywalk collapse in 1981, the Tacoma Narrows Bridge in 1940, the Liberty Bell in 1752, Grumman Flxible Buses in New York in the early 1980s, and early iron bridges of the mid nineteenth century. The author discusses some designs that took a risk and succeeded such as the Crystal Palace in 1851.

Not only are the details of all these structures and their success or failure interesting, but so are the rigorous investigations and design reviews expended upon them. For those structures that experienced failures due to inadequate design or poor quality of construction, the postmortem efforts to pin down the exact root causes give one much to think about in terms of software failures.

Rarely are failed software systems explored in public, the recent Denver airport baggage handling fiasco notwithstanding. The software development community could learn much if successful and unsuccessful systems were examined in great detail. But what company wants to open itself up to examination of its process for designing, developing, and deploying systems?

An interesting chapter toward the end of the book discusses the demise of the slide rule in favor of calculators and later computers and CAD software. Having used a slide rule in high school and switched to a calculator in college when they became somewhat affordable, I paid extra attention to this topic. The author goes into great detail discussing the limits in precision that the slide rule imposed upon designs, which prompted engineers to allow extra margin for errors. Overconfidence in the perceived accuracy of computer models, he argues, leads engineers to attempt more complex designs than in the day of the slide rule and to place too much trust in the correctness of the software and hence the design.

The author worries that engineers are no longer learning their craft properly because of technological advances. Because engineering is an endeavor to build safe structures in a cost-efficient manner, relying upon the integrity of the computer model becomes increasingly important. With increased numerical accuracy over the slide rule and the ability to perform complex calculations, the tendency might be to try to achieve an optimal design. The Hartford Civic Center roof collapse in January 1978 is discussed as an example of what can happen when we trust computer assisted-designs too much.

The work products of the structural engineer are constantly in the public eye. If a bridge, building, or other structure fails, everyone knows about it. It is important news. The public, government agencies, and the structural engineering profession demand to understand why such failures occur and to ensure that they don't happen again.

However, failure in the software industry not only is more prevalent than in structural engineering, but almost expected and too often tolerated. Although there are many aspects of software design and development (sorry, I can't call it engineering because it isn't) that are different from traditional engineering disciplines, certainly the entire field could prosper from an open analysis and disclosure of why software projects fail. Too often different efforts are failing for the same reason as earlier efforts. But there is little prior art to turn to.

The USENET Handbook: A User's Guide to Netnews

by Mark Harrison, O'Reilly & Associates, 1995,
ISBN: 1-56592-101-1, 388 pp., $24.95.

Reviewed by Rick Umali
<rgu@world.std.com>

For people who have access to UNIX, and have never read USENET News, I heartily recommend Mark Harrison's *The USENET Handbook: A User's Guide to Netnews*. For those who have access to UNIX and read USENET news, I still recommend this book.

A book about Netnews would have been somewhat laughable in "the old days." When I first started reading news in 1986, my only introduction was "type rn."But as a budding hacker, I soon learned the wily ways of USENET. I went through the phase of being a USENET junkie and eventually corralled my first "real" job through news (*ne.jobs*).

So it was with skepticism that I began reading Mark Harrison's book. What could this book teach me about Netnews that I didn't already know, usually from news itself?

It turns out that this book does teach a lot for all comers to Netnews. I was exposed to a lot of new ideas regarding "archiving" local news for training and tracking purposes (p. 181). I read about newsindex, a Perl script that creates a WAIS index for superior retrieval of archived articles.

Mark Harrison tours through the nn, tin, gnus, and trumpet newsreaders, which broadened my knowledge about other newsreaders (I first used rn and later graduated to trn).

Sheepishly, I realized how much I don't know about news: *comp.archives* was always a mystery to me until I read this book. The use of the distribution field was finally clear to me after reading the appendix on news distributions (p. 255).

For the experienced Netnews reader, the book contains lots of gems. There are classic postings from Gene Spafford ("It is both heartening and unfortunate that there are so many well-meaning people who continue to propagate these stories.") and Edward Vielmetti ("Usenet is a right, a left, a jab, and a sharp uppercut to the jaw. The postman hits! You have new mail."). There are references to Kibo (including a primer on kibology by Valerie Quercia), *net.suicide*, and *net.parties* by Clairemarie Fisher O'Leary.

It was rather strange reading some classic FAQs in a book (in nice type, no less!), instead of on the screen.

The real strength of this book is its clear explanation of hoary subjects like uudecoding, making shar files, *net.jargon*, ROT13, and smileys. Although you don't need a book to learn how to manipulate news, this book would most assuredly reduce the learning curve for new users to USENET News.

Mark Harrison recommends the newsgroup *news.announce. newusers* to everyone reading USENET news, which is certainly sage advice for the amateur and the pro Netnews reader.

There is also plenty of advice for posting news. If every newbie followed the clear advice of Chapter 7 "Posting Articles to the Net," USENET would have less inflammatory material.

The book is definitely UNIX-centric, which is only to be expected. The chapter entitled "Software, Pictures, and Other Goodies" has references to UNIX shell commands. Although there are nods to the PC DOS/Windows audience (*arc, pkunzip,* and even a chapter on the trumpet news-reader), this book won't be enough for a DOS/Windows user to fully utilize news. This would probably be a worthy subject for another book.

As Vielmetti writes in his FAQ (I'm paraphrasing): there is a certain culture about the net that has grown up on UNIX machines, which occasionally runs into fierce clashes with the culture that has grown up on IBM machines, C-64's, MS-DOS Fidonet systems, commercial chat systems, and "family oriented" systems.

Like people reading a travel book about your hometown or watching a movie that takes place there, experienced news readers will recognize many of the sights and attractions in this book. And like any well-written tour guide, it will make anyone want to visit USENET again.

Announcement and Preliminary Call for Papers   ▓SENIX

2nd Conference on Object-Oriented Technologies and Systems (COOTS)

June 17–21, 1996
Toronto, Canada
Sponsored by the USENIX Association

Important Dates

Tutorial submissions due: *Feb 7, 1996*

Paper submissions due: *Feb 13, 1996*

Notification to authors: *March 5, 1996*

Camera-ready final papers due: *May 7, 1996*

Preliminary Program Committee

Program Chair: Douglas C. Schmidt, *Washington University*

Tutorial Program Chair: Doug Lea, *SUNY Oswego*

Donald Box, *Developmentor*
Kraig Brockschmidt, *Microsoft*
David Chappell, *Chappell and Associates*
Andrew Chien, *University of Illinois, Urbana-Champaign*
David Cohn, *University of Notre Dame*
Jim Coplien, *AT&T Bell Labs*
Murthy Devarokonda, *IBM Watson Research Labs*
Peter Druschel, *Rice University*
Daniel Edelson, *IA Corporation*
Nayeem Islam, *IBM Watson Research Labs*
Dennis Kafura, *Virginia Tech University*
Doug Lea, *SUNY Ostwego*
Dmitry Lenkov, *Hewlett-Packard*
Mark Linton, *Silicon Graphics Inc.*
Vince Russo, *Purdue University*
Jerry Schwarz, *Declarative Systems*
Kevin Shank, *Rochester Institute of Technology*
Michael Stal, *Siemens AG*
Bjarne Stroustrup, *AT&T Bell Labs*
Steve Vinoski, *Hewlett-Packard*
Jim Waldo, *Sun Microsystems Labs*

Overview

The COOTS conference is intended to showcase advanced R&D work in object-oriented technologies and software systems. The conference emphasizes experimental research and experience gain by using object-oriented techniques and languages to build complex software systems that meet real world needs.

Tutorials

The COOTS conference will begin with two days of tutorials. The tutorial program will offer a selection of tutorials from among several tracks. We expect tutorial topics to include:

Distributed object systems (CORBA, Network OLE, DSOM, etc.)

C++ Standard Template Library

Object-oriented network programming

Design patterns for object-oriented systems

Evolution of ANSI/ISO C++ standardization

Concurrent object-oriented programming

Efficient and effective framework design

Alternative object-oriented languages

Tutorial proposal submissions must be received by February 7, 1996. The preferred form of submission is via electronic mail to the Tutorial Chair Doug Lea *(dl@g.oswego.edu)*. Tutorials selected for presentation at the conference will be announced by February 20, 1996.

Technical Session Topics

Two days of technical sessions will follow the tutorials. We seek papers describing original work concerning the design, implementation, experimentation, and use of object-oriented technologies. Like the USENIX C++ conferences from which it is derived, COOTS emphasizes advanced engineering aspects of object technology, focusing on experimental systems research and development on distributed objects, multimedia, operating systems, compiler technology, and C++. While papers covering work in C++ are strongly encouraged, the conference is broader in scope than its ancestors. In particular, we invite submissions describing results and work in other object-oriented or object-based languages.

Potential topics include but are not limited to:

Applications of, and experiences with, object-oriented technologies in various domains (distributed systems, multimedia, real-time systems, financial services, human/computer interface, etc.)

Distributed object systems (CORBA, DCOM, DSOM, etc.)

Implementations of commercial object infrastructures and reliable distributed objects (CORBA, NextStep, OLE/COM, SOM, Isis/RDO, etc.)

C++ standardization (STL, templates, implementation challenges)

Object-oriented programming language
development environments and tools
(C++, Modula-3, Eiffel, etc.)

Content-oriented languages for
programming in the WWW (Java,
Python, Obliq, Phantom, etc.)

Interface description languages (DCE
IDL, OMG IDL, etc.)

Questions regarding a topic's relevance to
the conference may be addressed to the
program chair via electronic mail to
schmidt@cs.wustl.edu. Proceedings of the
conference will be published by USENIX
and will be provided free to technical
session attendees; additional copies will
be available for purchase from USENIX.

In addition, based upon feedback solic-
ited at the conference from attendees, the
program committee will select five papers
to be published in revised and expanded
form in a special issue of a suitable jour-
nal. To help authors prepare these papers
for publication, we will have one or more
BOF sessions organized as "writers work-
shops." The writers workshop format has
a group of "discussants" read the paper
carefully before the session. During the
writers workshop, the discussants
examine the strengths and weaknesses of
each paper, accentuating positive aspects
and suggesting improvements in content
and style. The author is "invisible" during
this discussion, and is expected to take
notes and revise the paper in accordance
with the comments.

Advanced Topics Workshop

This year's USENIX COOTS conference
will conclude with an Advanced Topics
Workshop. The goal of this workshop is to
provide an informal setting in which to
exchange in-depth technical information
with your peers. This workshop will be
open to authors of papers in the confer-
ence, as well as participants who submit
position papers related to the workshop's
topic. This topic will be determined
several months before the conference
and a Call for Position papers will be

announced. Past USENIX C++ confer-
ences have held Advanced Topics Work-
shops on a variety of topics including
distributed object computing and imple-
mentation issues related to C++ language
features.

What to Submit

Technical paper submissions must be
received by February 13, 1996. Full papers
should be 10 to 15 pages (around 5,000–
6,000 words). In lieu of a full paper,
authors may submit extended abstracts
that discuss key ideas. Extended abstracts
should be 5–7 pages long (about 2,500–
3,500 words), not counting references
and figures. The body of the extended
abstract should be written as complete
paragraphs. The objective of an extended
abstract should be to convince reviewers
that a good, solid paper and presentation
will result. Extended abstracts are intend
to stimulate industrial participation and to
allow publication of very current material.

All submissions will be judged on origi-
nality, relevance, and correctness. Each
accepted submission will be assigned a
member of the program committee to act
as its shepherd through the preparation of
the final paper. The assigned member will
act as a conduit for feedback from the
committee to the authors. Camera-ready
final papers are due May 7, 1996.

Please accompany each submission by a
cover letter stating the paper title and
authors, along with the name of the person
who will act as the contact to the program
committee. Please include a surface mail
address, daytime and evening phone
number, and, if available, an email address
and fax number for the contact person.

If you would like to receive detailed
guidelines for submission and examples
of extended abstracts, you may telephone
the USENIX Association office at:
510.528.8649,
or email to
cootsauthors@usenix.org
or to the program committee chair
schmidt@cs.wustl.edu

An electronic version of this Call for
Papers is available on the World Wide
Web. The URL is: *http://www.usenix.org*.

The COOTS conference, like most
conferences and journals, requires that
papers not be submitted simultaneously to
another conference or publication and
that submitted papers not be previously
or subsequently published elsewhere.
Papers accompanied by non-disclosure
agreement forms are not acceptable and
will be returned to the author(s) unread.
All submissions are held in the highest
confidentiality prior to publication in the
Proceedings, both as a matter of policy
and in accord with the U.S. Copyright
Act of 1976.

Where to submit

Please send one copy of a full paper or an
extended abstract to the program commit-
tee via one of the following methods. All
submissions will be acknowledged.

- Preferred Method: email
  (Postscript or ASCII) to

  cootspapers@usenix.org

- Alternate Method: postal delivery to

  USENIX COOTS Conference
  c/o Dr. Douglas C. Schmidt
  Department of Computer Science
  Washington University
  Campus Box 1045
  One Brookings Drive
  St. Louis, Missouri 63130-4899
  U.S.A.
  (TEL): 314.935.7538
  (FAX): 314.935.7302

Registration Materials

Materials containing all details of the
technical and tutorial programs, registra-
tion fees and forms, and hotel information
will be available beginning in April 1996.
If you wish to receive the registration
materials, please contact USENIX at:

USENIX Conference Office
22672 Lambert St., Suite 613
Lake Forest, CA USA 92630
Tel: 714.588.8649
Fax: 714.588.9706
Email: *conference@usenix.org*

4th Annual Tcl/Tk Workshop '96

Sponsored by the USENIX Association
July 10–13, 1996 in Monterey, California

The fourth annual Tcl/Tk workshop, sponsored by the USENIX Association, will be held July 10-13, 1996 in Monterey, California. The workshop is a forum to:

Bring together Tcl/Tk researchers and practitioners.

Publish and present current work.

Plan for future Tcl/Tk related developments.

The workshop program will include formal presentations of papers and demonstrations, as well as informal demonstrations, work in progress sessions, birds of a feather sessions, and tutorials.

This call provides information on submitting formal papers and demonstrations. Information on registration will be available separately in April, 1996.

Structure of Submissions

Papers and demonstrations should report on original Tcl/Tk research. Example topics have included system extensions, novel Tcl/Tk based applications, reports on experiences building particular applications, use of different programming paradigms within Tcl, and proposals for new directions. All work must be original, and not submitted elsewhere. The audience for the workshop is researchers and practitioners who are expert users of Tcl/Tk.

There are three types of submissions: applications papers, general papers, and demonstrations. Both paper categories are limited to ten pages, and authors will be given a twenty minute time slot at the workshop. A full version of the paper must be submitted for review. Live demonstrations of software will be given a thirty minute time slot at the workshop, and a paper of up to four pages must accompany the demonstration. Demonstrations are intended as a forum to highlight

and describe innovative technologies having a highly visual or interactive component; they are not intended as a forum for marketing-oriented presentations. Detailed instructions on submission format appear below.

Applications papers have typically proven difficult to write. Authors considering submission of these types of papers are encouraged to consider the following common causes of rejection:

Paper is blatant marketing material for commercial product.

Insufficient background on application domain so that the audience cannot follow; excessive use of domain specific buzzwords.

Too much information on the application, but not enough on the relevance of Tcl/Tk to the application.

Too little consideration of how the Tcl/Tk community could benefit from experiences; limited generalizability.

Application only illustrates a routine usage of Tcl/Tk.

Detailed Submission Instructions

We are accepting workshop submissions electronically, via email. Submissions should be sent as both plain text (with no extra markup), and as PostScript (formatted for an 8.5 × 11 page). When submitting PostScript, please strive to ensure that your file can be printed on a variety of printers. If accepted, both electronic and camera-ready hardcopy of the final version will be required.

Applications papers and general papers must be full length versions, and not just abstracts. Papers may be a maximum of ten pages in length. If accepted, we would encourage use of brief video clips or demos during the presentation. If you think you may use AV equipment other than standard overheads or

35mm slides, please make a note of it on the cover sheet described below. This information is not used to judge your submission, but will assist in organizing the final program.

Submissions for demonstrations must prepare a paper of up to four pages in length describing the demonstration and providing further background; this will appear in the workshop proceedings. Submitters are encouraged to submit additional material supporting their demonstration for review, such as a storyboard or outline. If you would like to submit other material, such as videotapes, contact the conference chairs for more information.

This workshop, like most conferences and journals, requires that papers not be submitted simultaneously to another conference or publication and that submitted papers not be previously or subsequently published elsewhere. Papers accompanied by "non-disclosure agreement" forms are not acceptable and will be returned to the author(s) unread. All submissions are held in the highest confidentiality prior to publication in the Proceedings, both as a matter of policy and in accord with the U.S. Copyright Act of 1976.

A cover letter should be included with all submissions containing the following information:

- Name of all authors
- Primary contact
- Full postal address
- Telephone number
- Email address (very important)
- Category (application paper, general paper, or demonstration)
- Anticipated AV needs (used only for planning purposes)

Submissions should consist of a uuencoded, compressed tar file (compress or gzip), containing both the plain text and PostScript versions (filenames should be based on your last name, e.g. smith.txt and smith.ps). This should be mailed, along with the cover letter, to *tcl96@sco.com*. Receipt of submissions will be acknowledged by return email within one week. If an acknowledgement is not received, please contact the co-chairs listed below.

Important Dates

Deadlines for paper and demo submissions: *March 5, 1996*

Notification of acceptance: *April 16, 1996*

Camera-ready copy due: *May 28, 1996*

Program Committee

Co-chairs:

Mark Diekhans, *SCO*
  markd@sco.com

Mark Roseman, *University of Calgary*
  roseman@cpsc.ucalgary.ca

Program Committee:

Ben Bederson, *University of New Mexico*

Wayne Christopher, *ICEM CFD Engineering*

Joe Konstan, *University of Minnesota*

Don Libes, *NIST*

Michael McLennan, *AT&T*

Larry Rowe, *University of California Berkeley*

Brent Welch, *Sun Microsystems Laboratories*

Will Wilbrink, *Unisys Canada*

David Young, *SCO*

More information on the workshop will be posted to *comp.lang.tcl, comp.org.usenix*, and placed on the World Wide Web at *http://www.usenix.org* as it becomes available.

Registration Materials

Materials containing all details of the technical and tutorial programs, registration fees and forms, and hotel information will be available in April, 1996. If you wish to receive the registration materials, please contact:

USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest CA 92630
714.588.8649
Fax: 714.588.9706
Email: *conference@usenix.org*
URL: *http://www.usenix.org*

Announcement and Preliminary Call for Papers **USENIX**

6th UNIX Security Symposium
Focusing on Applications of Cryptography

July 22–25, 1996
Fairmont Hotel, San Jose, California

Sponsored by the USENIX Association, the UNIX and Advanced Computing Systems Professional and Technical Association

Co-sponsored by UniForum

In cooperation with The Computer Emergency Response Team (CERT), and IFIP WG 11.4

Important Dates

Dates for Refereed paper submissions:
 Extended abstracts due:
  March 19, 1996
 Program Committee decisions made:
  April 15, 1996
 Camera-ready final papers due:
  June 10, 1996

Registration Materials Available:
 End of April 1996

Program Committee

Program Chair: Greg Rose, *Sterling Software*

Fred Avolio, *Trusted Information Systems, Inc.*
Steve Bellovin, *AT&T Bell Laboratories*
Brent Chapman, *Great Circle Associates*
Diane Coe, *The MITRE Corporation*
Ed DeHart, *CERT*
Kathy Fithen, *CERT*
Dan Geer, *Open Market Inc.*
Peter Gutmann, *University of Auckland*
Kent Landfield, *Sterling Software*
Clifford Neuman, *University of Southern California*
Avi Rubin, *Bellcore*
Eugene Spafford, *COAST Laboratory, Purdue University*
Ken van Wyk, *Defense Information Systems Agency*
Karen Worstell, *The Boeing Company*

Readers: Matt Bishop, *U.C. Davis;*
 Lee Damon, *Qualcomm;*
 Phil Karn, *Qualcomm*

Overview

The goal of this symposium is to bring together security and cryptography practitioners, researchers, system administrators, systems programmers, and others with an interest in applying cryptography, network and computer security, and especially the area where these overlap. The focus on applications of cryptography is intended to attract papers in the fields of electronic commerce and information processing, as well as security. Please note that papers about new cryptographic algorithms are not solicited; however new applications are.

This will be a four day, single track symposium with tutorials, refereed and technical presentations, and panel discussions. Tutorials will take place the first two days followed by two days of technical sessions.

Tutorials
July 22-23

Tutorials for both technical staff and managers will provide immediately useful, practical information on topics such as local and network security precautions, what cryptography can and cannot do, security mechanisms and policies, firewalls and monitoring systems.

Technical Sessions
July 24-25

In addition to the keynote presentation, the technical program includes refereed papers and invited talks. There may be panel sessions. There will be Birds-of-a-Feather sessions and Works-in-Progress Reports on two evenings. You are invited to make suggestions to the program committee via email to

 security@usenix.org.

Papers that have been formally reviewed and accepted will be presented during the symposium and published in the symposium proceedings. Proceedings of the symposium will be

published by USENIX and will be provided free to technical session attendees; additional copies will be available for purchase from USENIX.

Symposium Topics

Presentations are being solicited in areas including but not limited to:

Anonymous transactions

Applications of cryptographic techniques

Attacks against secure networks/machines

Cryptanalysis and codebreaking as attacks

Cryptographic tools

Electronic commerce security

Firewalls and firewall toolkits

Legislative and legal issues

Case studies

Computer misuse and anomaly detection

File and File system security

Network security

Security and system management

Security in heterogeneous environments

Security incident investigation and response

Security tools

User/system authentication

Penetration testing

Malicious code analysis

Note that this symposium is not about new codes or ciphers, or cryptanalysis for its own sake.

How to Submit a Refereed Paper

Submissions must be received by March 19, 1996. Authors are encouraged to submit an extended abstract which discusses key ideas and demonstrates the structure of the finished paper. Extended abstracts should be 3-5 pages long (about 1500-2500 words), not counting references and figures. The body of the extended abstract should be in complete paragraphs. The object of an extended abstract is to convince the reviewers that a good paper and presentation will result. Full papers can be submitted if they are complete in advance of the date. Full papers should be 8 to 15 typeset pages.

Authors will be notified of acceptance on April 15, 1996.

All submissions will be judged on originality, relevance, and correctness. Each accepted submission will be assigned a member of the program committee to act as its shepherd through the preparation of the final paper. The assigned member will act as a conduit for feedback from the committee to the authors. Camera-ready final papers are due June 10, 1996.

Please accompany each submission by a cover letter stating the paper title and authors along with the name of the person who will act as the contact to the program committee. Please include a surface mail address, daytime and evening phone number, and, if available, an email address and fax number for the contact person.

If you would like to receive detailed guidelines for submission and examples of extended abstracts, you may send email to:

securityauthors@usenix.org

or telephone the USENIX Association office at 510.528.8649.

The UNIX Security Symposium, like most conferences and journals, requires that papers not be submitted simultaneously to another conference or publication and that submitted papers not be previously or subsequently published elsewhere. Papers accompanied by "non-disclosure agreement" forms are not acceptable and will be returned to the author(s) unread. All submissions are held in the highest confidentiality prior to publication in the Proceedings, both as a matter of policy and in accord with the U.S. Copyright Act of 1976.

Where to Submit

Please send one copy of an extended abstract or a full paper to the program committee via each of two, for reliability, of the following methods. All submissions will be acknowledged.

- **Preferred Method** email (Postscript or ASCII) to:

  securitypapers@usenix.org

- **Alternate Method** postal delivery to:

  Security Symposium
  USENIX
  2560 Ninth St., Suite 215
  Berkeley CA 94710
  U.S.A.
  Phone: 510.528.8649

- Fax: 510.548.5738

Registration Materials

Materials containing all details of the technical and tutorial programs, registration fees and forms, and hotel information will be available at the end of April 1996. If you wish to receive the registration materials, please contact USENIX at:

USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest, CA USA 92630
714.588.8649
Fax: 714.588.9706
email: *conference@usenix.org*

Information can also be found under the USENIX Association Web page, URL: *http://www.usenix.org*

Announcement and Preliminary Call for Papers **USENIX**

2nd Symposium on Operating Systems Design and Implementation (OSDI'96)

October 28–October 31, 1996
Seattle, Washington, USA

Sponsored by the USENIX Association
Co-sponsored by ACM SIGOPS and IEEE TCOS

After a successful first OSDI symposium, the next OSDI will continue to focus on practical issues related to modern operating systems. OSDI brings together professionals from academic and industrial backgrounds, and has become the perfect forum for issues concerning the design and implementation of operating systems for modern computing platforms such as workstations, parallel architectures, mobile computers, and high speed networks.

The OSDI symposium emphasizes both *innovative research* and *quantified experience* in operating systems. We seek papers describing original work concerning the design, implementation, and use of modern operating systems. Besides mature work, we encourage submissions describing exceptionally promising, well-grounded speculative work, or enlightening negative results. Topics of interest include, but are not limited to:

OS structure and organization
OS kernel internals, servers and applications
Distributed and mobile computing
Multiprocessor and parallel systems
Communications
Storage Management and I/O systems
Security in distributed systems
Scalability and availability
Heterogeneous systems
Performance and optimizations
Language support for OS
OS interaction with HW architecture
OS support for embedded systems
OS support for real time and multimedia
Interaction of OS and applications

Symposium Overview

The symposium will consist of one day of tutorials, followed by 2.5 days of single-track technical sessions with presentations of the refereed papers, and a half day workshop on a topic yet to be determined. One of the technical sessions will be dedicated to work-in-progress presentations and will be described in later announcements. The refereed papers will be published in the Proceedings, provided free to technical session attendees and available for purchase from USENIX. The Proceedings may also be distributed to ACM SIGOPS members. Papers of particular merit will be selected to receive an award and will be published in the IEEE TCOS Bulletin.

Program Committee

Karin Petersen, *Xerox PARC (co-chair)*
Willy Zwaenepoel, *Rice University (co-chair)*
Peter Chen, *University of Michigan*
Richard Draves, *Microsoft Research*
Carla Ellis, *Duke University*
Ed Felten, *Princeton University*
Jim Gray, *Microsoft Bay Area Laboratory*
Kevin Jeffay, *University of North Carolina*
David Johnson, *Carnegie Mellon University*
Jay Lepreau, *University of Utah*
Jeff Mogul, *DEC WRL*
Marc Shapiro, *INRIA*
John Wilkes, *Hewlett-Packard Labs*
John Zahorjan, *University of Washington*

Important Dates

Full papers due: *May 7, 1996*

Notification to authors: *July 30, 1996*

Revised papers due for shepherding: *August 19, 1996*

Camera-ready full papers due: *September 16, 1996*

Submission Process

Authors are required to submit full papers by May 7, 1996. Submitted papers should be no longer than 14 pages, spaced no closer than standard 10 point font on 12 point baseline, single- or double-column format. Longer submissions will be discarded without review. Very similar papers must not have been published or submitted for publication elsewhere. All submissions will be held in the highest confidentiality prior to publication. Papers accompanied by so-called "non-disclosure agreement" forms are not acceptable and will be returned unread.

The papers will be judged on significance, originality, clarity, relevance, and correctness. The committee will favor papers with reproducible results, especially those supplying detailed data and explanations, or offering to make data sets or source code available.

Accepted papers will be shepherded through an editorial review process by a member of the program committee.

Authors of accepted papers will be expected to provide an HTML page containing the abstract and links to their paper, slides, and software, if available. This will be collected after the event for inclusion in an electronic version of the symposium (for an example, see *http://www.cs.utah.edu/~lepreau/osdi94/*).

Where to submit

Submission of all papers must be made in both paper and electronic form. Fifteen (15) paper copies (double sided if possible) of the paper must be sent to:

Willy Zwaenepoel
Department of Computer Science,
Rice University
6100 S. Main St.
Houston, TX 77005, USA

and one electronic copy in Postscript (not ASCII) must be submitted by electronic mail to: *osdi-papers@cs.rice.edu*

For administrative reasons (not blind reviewing), every submission (in both its paper and electronic form) should include one additional page containing: (i) paper title and authors, indicating any who are full time students, and (ii)

for the author who will act as the contact to the program committee, his or her name, paper mail address, daytime and evening phone numbers, email address and fax number, if available. The cover sheet mailed with the electronic paper submission should be in ASCII to facilitate accurate on-line bookkeeping, and should be included in the same electronic mail message as the PostScript file containing the paper.

All submissions will be acknowledged by May 21, 1996. If your submission is not acknowledged by this date, please contact the program chairs promptly at *osdi@cs.rice.edu*.

Registration Materials

Materials containing all details of the technical and tutorial programs, registration fees and forms, and hotel information will be mailed in August 1996. If you wish to receive the registration materials, please contact:

USENIX Conference Office
22672 Lambert St., Suite 613
Lake Forest, CA USA 92630
Phone: 714 588 8649
Fax: 714 588 9706
Email: *conference@usenix.org*
WWW URL: *http://www.usenix.org.*

IEEE COMPUTER SOCIETY

Announcement and Call for Participation █SENIX█

10th Systems Administration Conference (LISA '96)

September 30–October 4, 1996
Chicago Marriott, Chicago, Illinois

Co-sponsored by USENIX, the Advanced Computing Systems
Professional and Technical Association and SAGE, the System Administrators Guild

Important Dates

Refereed paper submissions:
   Extended abstracts due: *May 7, 1996*
   Notification to authors by:
   June 11, 1996
   Final papers due: *August 15, 1996*
Registration materials available:
   July, 1996

Overview

LISA, the USENIX Systems Administration Conference, is the leading conference for and by system administrators. LISA originally stood for "Large Installation Systems Administration" when a large installation meant over 100 users, 100 systems, or a gigabyte of disk storage. Today, the LISA conference offers the most comprehensive program for system administrators from sites of all sizes and at all levels of experience.

   LISA '96 will mark the tenth anniversary of the LISA conference. While there will be events at the conference to mark this occasion, the focus will continue to be bringing system administrators the latest tools, techniques, and information needed to keep apace with the rapid technology advancements, changes in public and legal policy, and changes in the ways that their employers do business.

Tutorial Program

Monday and Tuesday,
September 30–October 1, 1996

The conference will offer up to 20 tutorials on two days. Tutorials are offered on all levels of system administration from novice to senior system administrator.

To provide the best possible tutorial offerings, USENIX continually solicits proposals for new tutorials. If you are interested in presenting a tutorial at this or other USENIX conferences, please contact the tutorial coordinator:

   Daniel V. Klein
   412.421.0285
   Fax: 412.421.2332
   Email: *dvk@usenix.org*

Technical Sessions

Wednesday through Friday,
October 2–4, 1996

The three days of technical sessions consist of two parallel tracks. The first track is dedicated to presentations of refereed technical papers. The second track is intended to accommodate invited talks, panels and Works-in-Progress (WIP) sessions.

Conference Topics

Papers addressing the following topics are particularly timely; papers addressing other technical areas of general interest are equally welcome.

- Innovative system administration tools and techniques
- Integrating new networking technologies
- Problem tracking
- Remote site administration
- Experiences supporting large sites (>1000 users or machines)
- Experiences supporting nomadic and wireless computing
- Integration of heterogeneous platforms— multiple UNIX platforms, PC/Mac integration, interfacing with legacy systems

- Integration of emerging technologies to system administration
- Support stratgies in use at your site
- Distributed system administration
- Proactive problem management
- OS/Platform migration strategies
- Performance analysis and monitoring
- Data management
- Security
- Ethics
- Asset management
- Training the user
- Incorporating commercial system administration technology for your site

Refereed Paper Submissions

An extended abstract is required for the paper selection process. Full papers are not acceptable at this stage; if you send a full paper, you must also include an extended abstract. "Extended" means 2–5 pages.

   Include references to establish that you are familiar with related work, and, where possible, provide detailed performance data to establish that you have a working implementation or measurement tool.

   Submissions will be judged on the quality of the written submission, and whether or not the work advances the state of the art of system administration. For more detailed author instructions and a sample extended abstract, send email to

   lisa10author@usenix.org
or call USENIX at 510.528.8649.

   Note that LISA, like most conferences and journals, requires that papers not be submitted simultaneously to more than one conference or publication and that submitted papers not be previously or subsequently

USENIX ⁱ, the Advanced Computing Systems Professional and Technical Association.

58 *;login:* VOL. 21 · NO. 1

published elsewhere. Papers accompanied by "non-disclosure agreement" forms are not acceptable and will be returned unread. All submissions are held in the highest confidence prior to publication in the conference proceedings, both as a matter of policy and as protected by the U.S. Copyright Act of 1976.

Authors of an accepted paper must provide a final paper for publication in the conference proceedings. At least one author of each accepted paper presents the paper at the conference. Final papers are limited to 20 pages, including diagrams, figures and appendixes, and must be in troff, ASCII, or LaTeX format. We will supply you with instructions. Papers should include a brief description of the site, where appropriate.

Conference proceedings, containing all refereed papers and materials from the invited talks, will be distributed to attendees and will also be available from USENIX following the conference.

Where to Send Submissions

Please submit extended abstracts for the refereed paper track by two of the following methods:

Email to: *lisa10papers@usenix.org*
Fax to: 510.548.5738

Mail to:
LISA 10 Conference
USENIX Association
2560 Ninth Street, Suite 215,
Berkeley, CA USA 94710

To discuss potential submissions and for inquiries regarding the content of the conference program, contact the program co-chairs at *lisa10chair@usenix.org* or at:

Helen E. Harrison
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
919.677.8000 x6981
Fax: 919.677.4444
Email: *helen@usenix.org*

Amy K. Kreiling
Campus Box #3175
Department of Computer Science
University of North Carolina
Chapel Hill, NC 27599
919.962.1843
Fax: 919.962.1799
Email: *amy@usenix.org*

Invited Talk Track

If you have a topic of general interest to system administrators, but that is not suited for a traditional technical paper submission, please submit a proposal for a second track presentation to the invited talk (IT) coordinators at *itlisa@usenix.org*.

Workshop: Advanced Topics in System Administration
Tuesday, Oct 1, 1996

A one-day, pre-LISA conference workshop organized by John Schimmel of Silicon Graphics will focus on a discussion of the latest-breaking technical issues in the systems administration arena as introduced by those in attendance. Attendance is limited and based on acceptance of a position paper. A representative subset of positions will be discussed in an open forum.

How to Submit: Potential workshop attendees are invited to submit a proposal of at most 3 pages (ASCII) via electronic mail to *jes@sgi.com* no later than August 1. (More substantive reports of completed works should instead be submitted as papers to the technical sessions.) These proposals should briefly contain a topic for discussion, a description of the subject, an explanation of what makes this topic controversial or interesting, and a personal position. Acceptance notices to all participants will be issued by September 9, 1996.

Participants must be pre-registered for the LISA conference. No additional fee will be charged to attend this workshop, and lunch will be provided.

Program Committee

Program Co-chairs:
Helen E. Harrison, *SAS Institute Inc.*
Amy K. Kreiling, *University of North Carolina*

Program Committee:
Paul Evans, *Synopsys, Inc.*
David L. Kensiski, *MCI Telecommunications*
Bill LeFebvre, *Argonne National Labs*
E. Scott Menter, *Enterprise Systems Management*
Pat Parseghian, *AT&T Bell Laboratories*
Pat Wilson, *Dartmouth College*
Elizabeth Zwicky, *Silicon Graphics, Inc.*

Invited Talks Co-ordinators:
Rik Farrow, *Internet Security Consulting*
Kimberly Trudel, *Massachusetts Institute of Technology*

Vendor Displays
Wednesday and Thursday, October 2–3, 1996

LISA attendees have an enormous interest in industrial strength, state of the art solutions to system adminstration problems. If your company's products provide solutions, LISA will provide attendees with the technical expertise to understand and appreciate it. Please contact:

Zanna Knight
Tel: 510.528.8649
Fax: 510.548.5738
Email: *display@usenix.org*

Birds-Of-A-Feather Sessions

Birds-of-a-Feather sessions (BoFs) are very informal gatherings of attendees interested in a particular topic. BoFs are held Tuesday, Wednesday, and Thursday evenings of the conference. BoFs may be scheduled in advance by telephoning the USENIX Conference Office at 714.588.8649 or via email to *conference@usenix.org*. They may also be scheduled at the conference.

For Registration Information

The complete program and registration information will be available in July 1996. If you would like to receive registration materials, please contact:

USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest, CA 92630
Phone: 714.588.8649
Fax: 714.588.9706.
Email: *conference@usenix.org*.
URL: http://www.usenix.org

Or you can send email to our mailserver at *info@usenix.org*. Your message should contain the line: send catalog. A catalog will be returned to you.

Fifth International Workshop on Object-Orientation in Operating Systems: IWOOOS '96

October 27-28, 1996 – Seattle, WA

Sponsored by the IEEE Technical Committee on Operating Systems and Application Environments (TCOS) (pending) and USENIX Association

The fifth International Workshop on Object-Orientation in Operating Systems will bring together researchers and practitioners who are interested in object-oriented approaches to operating systems design, development, and application support. The purpose of the workshop is to provide an informal format and atmosphere in which ideas and current work can be presented and discussed at length. The workshop is designed to encourage the full participation of each attendee: both presenters and participants will be active contributors throughout the workshop.

This year's workshop will be held in Seattle, Washington, just prior to the Second Symposium on Operating Systems Design and Implementation which will be held in Seattle, Washington from October 28-31. We hope that the conjunction of the two events will foster cross-fertilization between related research communities.

The focus of the workshop is on how to effectively use objects inside operating systems and how to provide system support for object-oriented applications in a variety of application domains.

Subjects of particular interest include:

- Using objects to make operating systems customizable, extensible and adaptable

- Design patterns in operating systems

- Objects on WWW and their OS support

- Persistent objects and their OS support

- Mobile Objects and their OS support

The workshop is structured to encourage the submission of explorative work in the form of position papers. Position papers should be a maximum of 2500 words and should present initial work, new ideas, or a strong position statement.

Attendance will be by invitation only. To be invited, an attendee must submit a position paper. All submissions will be reviewed. All accepted papers will be published in a proceedings. The official language for the conference will be English.

Organizing Committee

Workshop Chair: Andrew Black, *Oregon Graduate Institute*
Program Chair: Nayeem Islam, *IBM T. J. Watson Research Center*
Local Arrangement Co-Chairs: Michael Jone, *Microsoft* and Crispin Cowan, *Oregon Graduate Institute*
Publicity Chair: Douglas Schmidt, *Washington University, St. Louis*
Publication Chair: Luis-Felipe Cabrera, *IBM*
Finance Chair: David Cohn, *University of Notre Dame*

Program Committee

Mustaque Ahamad, *Georgia Institute of Technology*
Henri Bal, *Vrieje University*
Gary Lindstrom, *University of Utah*
Eric Manning, *University of Victoria*
Satoshi Matsuoka, *University of Tokyo*
Gregor Kiczales, *Xerox Parc*
Sacha Krakowiak, *IMAG, France*
Jim Purtilo, *University of Maryland*
John Rosenberg, *University of Sydney*
Margo Seltzer, *Harvard University*
Santosh Shrivastava, *University of Newcastle-upon-Tyne*
Mario Tokoro, *Keio University*

Important Dates

Position Papers: July 1, 1996
Invitations issued: July 30, 1996
Camera-ready copy due: September 1, 1996

Submission Instructions

Each submission should have a principal author, to whom all messages will be sent; please provide email and postal addresses as well as telephone and fax numbers. A notice will be sent to the principal author upon receipt of every paper.

Electronic submission via email is strongly encouraged. Please send your paper to the program chair at *nayeem@watson.ibm.com*

Submissions are required to be in HTML, ASCII, or Post-Script (uuencoded). Electronic submissions will be ACK'ed within a day or so of receipt. If the submission could not be successfully printed out on paper then the program chair will attempt to confer with the sender via email about what to do as an alternative submission means. Note: if you do not receive any acknowledgment message at all within a period of several days then the submission message may have gotten lost in transit and you should send a short email message to the program chair to alert him to the difficulty.

Network Security'96

Pre-Announcement and Call for Participation

The Second Annual Conference on UNIX Network Security

November 4 - 8, 1996
Washington, D.C.

Sponsored by The Network Security Institute.
In cooperation FedUNIX, and the Escal Institute.

Important Dates	
Extended abstracts due:	June 14th, 1996
Notification to authors:	July 125th, 1996
Camera-ready papers due:	September 20th, 1996

The annual Network Security conference is an off-shoot of the very successful SANS (System Administration, Networking, and Security) Conferences. The first annual Network Security conference was very successful, with over 300 attendees and featured such notable speakers such as Matt Bishop, Marcus Ranum, Gene Schultz and Bill Cheswick. Network Security'96 is expected to be larger and will include three days of in-depth, authoritative courses and two days of multiple parallel tracks comprising refereed paper presentations on real-world case studies, invited technical presentations, and panel sessions. The Network Security Fall conference provides a forum at which UNIX security professionals can exchange practical information, share new ideas, evaluate new tools and, most importantly, expand their network of professional contacts.

Some of the comments from Network Security'95 attendees include:

"It was GREAT!" (Ken Chan, SIAC)

"Worth the Journey." (Van Vr. Zuren, Sun Belgium)

"Exactly what I was looking for: closing the gap between technical and organizational aspects, giving the helicopter view and a useful tool (the poster)." (Magda deJong, H-P Netherlands)

"Equivalent to months of work scouring the network archives looking for information-- a well presented summary of tools and vulnerabilities." (Chris Gorsuch, Texas Instruments)

"The information was invaluable!" (Hal Lewis, USAISC)

"I attended all 5 days, and the quality was so high that I never reached the burnout or saturation (or hookey) stage." (Ken Eichamn, CAS)

Network Security'96 continues the tradition of focusing exclusively on practical solutions to today's security problems. It brings together in one place, the top experts, the right technical sessions and the key courses all presented by the industry's most effective speakers. Network Security'96 will cover such topics as advanced firewall design, intrusion detection systems, network encryption alternatives and virtual private networks. The refereed technical presentations will emphasize real-world case studies and panel sessions. Evenings will include both Birds Of a Feather sessions and other special events.

The Network Security program committee is seeking submissions for tutorials and panel sessions in addition to technical presentations. Tutorial topics in the past have included: firewalls, UNIX security tools, The Kerberos system, building a security infrastructure, and WEB security issues. Panel sessions of interest include those with controversial or alternative viewpoints as well as those that encourage enlightening discussion of relevant issues.

Formally reviewed papers, presented at the conference, will be published in the conference proceedings and provided to all attendees.

Potential Paper Topics

- Designing and implementing security policies (plus experiences)
- Detecting and responding to security incidents
- Experiences with intruders
- Experiences with intrusion detection systems
- Interpreting security incident forensics
- Managing security at large heterogeneous sites
- Managing security for mobile and remote sites
- Protecting data with innovative techniques and methods
- Securing your site as you join the Internet
- Securing a WWW machine (different from normal machines)
- Implementing authentication on your web site
- Developing secure web applications
- Using innovative security auditing and monitoring techniques
- Using one-time password systems and other authentication techniques

- Using new and radical methods for controlling network communication such as packet filters, firewalls, host-based filter schemes, and the new security routers
- Future trends in firewall design
- Future trends on security and the World Wide Web

Abstract Submissions
Extended abstracts for refereed papers must be 2 to 5 pages long; 1500-2000 words is a good length. The object of an extended abstract is to convince the reviewers that a good paper and presentation will result. Final papers should be 8 - 12 pages long. Time slots for the technical presentations will be 30 or 45 minutes. Please specify time needed.

Your abstract should include:
1. An author information block including the title of the paper, the principal author's name, address, email, telephone, and FAX numbers, and the names of the other authors.
2. A description of the problem and its importance.
3. The solution, including details of how it worked. If the work builds on emerging technology, try to show what the expected impact will be when the technology matures. If your solution is based on commercial hardware or software tools, name them. (Abstracts from software vendors are also welcome, and will be considered as part of the commercial tools track or the regular paper sessions, depending on their focus.)
4. Data on the solution's effectiveness: before and after comparisons, evaluations, direct savings, trade-offs, etc.
5. Lessons learned.

Panel proposals should be a minimum of one page and should include the topic(s) and objective(s) of the panel. Names of potential panelist members should also be included. Time slots for panel sessions will be either 45 or 90 minutes. Please specify time needed.

Tutorial proposals should be 3 to 5 pages long and should include: a full outline and/or a written description of the tutorial, the intended audience, the level of the material (beginning, intermediate or advanced), a minimum of two references for the author's tutorial ability and a brief biography (1 paragraph) for the author. Tutorials are either full-day (6 hours of class) or half-day (3 hours of class). Please specify time needed.

Best Stories Contest
Network Security'96 will continue the tradition of the "Best Stories" contest. Two people will win free attendance (i.e., registration fees will be waived) at the technical conference. All you need to do to enter is write a 500 to 1000 word description of a success or failure pertaining to UNIX system security. It doesn't have to be a funny story from the newspapers, just tell us a funny or practical or illuminating (or all three) true security story.

The stories will be graded according to their level of educational and entertainment value. Stories can be about "failures" as well as "successes". Good stories will fit the following outline:

1. What problem was being faced?
2. What tools or techniques were selected to approach the problem?
3. What happened when you tried to solve the problem?
4. What lessons were learned?
5. What would you do differently if you had to do it over again.

The two winners will present their stories during a single session at the technical conference (November 7th or 8th).

Where To Submit
Please send one copy of your extended abstract to the program committee using one of the following methods. All submissions will be acknowledged.

Preferred method: email (plain ASCII text) to
sans@clark.net

Alternative method: postal delivery (on paper) to
Network Security Abstracts
4610 Tournay Road
Bethesda, MD 20816

Registration Materials
Materials containing all details of the Network Security'96 conference will be mailed in August 1996. If you would like to receive the registration materials, please contact:

Network Security Conference Office
4610 Tournay Road
Bethesda, MD 20816
Phone: 719-599-4303
FAX: 719-599-4394
Email: *sans@clark.net*

Conference Chairs
Matt Bishop, *University of California at Davis*
Michele Crabb, *Sterling Software at NASA Ames Research Center*
Alan Paller, *CIO Institute*

Program Committee
Ron Holland, *Jet Propulsion Laboratory*
Mike Parker (der Mouse), *McGill University in Montreal, Quebec*
Marcus Ranum, *Information Warehouse*

Announcement and Preliminary Call for Papers **USENIX**

USENIX Annual Technical Conference
Pre-Announcement and Call for Papers and Presenters

January 6–10, 1997
Anaheim Marriott Hotel
Anaheim, California

Program Committee

John T. Kohl, *Program Chair, Atria Software*
Matt Blaze, *AT&T Bell Research*
Nathaniel Borenstein, *First Virtual Holdings*
Charlie Briggs, *Digital Equipment Corporation*
Clem Cole, *Locus Computing*
Fred Douglis, *AT&T Bell Research*
Rob Gingell, *Sun Microsystems*
Mike Karels, *Berkeley Software Design*
Peg Schafer, *Harvard University*
John Schimmel, *Silicon Graphics*
Carl Staelin, *Hewlett-Packard Laboratories*

Due Dates for Refereed Paper Submissions

Manuscripts Due: June 18, 1996
Notification to Authors: August 7, 1996
Camera-ready Final Papers Due: November 13, 1996

Conference Schedule Overview

Tutorials: January 6-7, 1997
Technical Sessions and Invited Talks: January 8-10, 1997
Birds-of-a-Feather Sessions: January 7-9, 1997
Vendor Display: January 8-9, 1997
USENIX Reception: January 8, 1997

Conference Overview

The conference technical sessions include one track of refereed papers selected by the Program Committee. The refereed papers are published in the Conference Proceedings which are provided to all registered technical session attendees.

There is also a parallel track of Invited Talks. These survey-style sessions given by experts range over a variety of interesting and timely topics. Submitted Notes from the Invited Talks are published and distributed to registered technical sessions attendees.

Two full days of tutorials precede the technical sessions with practical tutorials on timely topics.

Other highlights of the conference include a work-in-progress session, which provides a forum for short informal technical presentations; the evening birds-of-a-feather sessions which provide very informal gatherings on particular topics; the Guru is IN sessions, informal discussions where noted experts from the USENIX community will answer technical questions; and vendor exhibits, which provide the opportunity for no-nonsense evaluation of products and services.

Refereed Papers

The emphasis for the 1997 USENIX Technical Conference is on advanced systems' uses in the global computing environment. How do we build computing systems which fulfill current needs yet can grow to handle the future demands? What techniques and tech-nologies can we use to satisfy a large, growing, and changing computing appetite? How do we support new computing styles with advanced computing systems? How do we protect the systems we build from failures or abuses?

The Program Committee seeks original and innovative full papers about the applications, architecture, implementation, and performance of modern computing systems. Some particularly interesting related topics are:

- Scaling the advanced system: down to laptops, palmtops, embedded systems; up to large file systems and memories, mass storage, faster networks, new protocols
- Mobile systems: network connectivity, system support, application design
- Tasks/roles where advanced systems shine or fall short
- Practical network security, privacy, and cryptography
- Electronic commerce, internetworking
- Multi-media challenges, solutions, and innovations
- Interoperation/standards: tools, techniques, and experience connecting with other computing systems

This list is by no means exhaustive; you are encouraged to submit papers on other advanced system related topics. As at all USENIX conferences, papers that analyze problem areas and draw important conclusions from practical experience are especially welcome.

How to Submit a Refereed Paper

It is imperative that you contact the USENIX Association office to receive

detailed guidelines and suggestions for submitting a quality paper to the refereed track of the technical sessions. Please send email to *usenix97authors@usenix.org* or telephone 510.528.8649.

In addition, specific questions about submissions to the USENIX 1997 Technical Conference may be sent to the program chairman via email at *kohl. @usenix.org*.

The program committee will review full papers this year (rather than extended abstracts as in the past). Papers should be 8 to 12 single-spaced 8.5"x11" pages (about 4000-6000 words), not counting figures and references. Papers longer than 12 pages will be discarded without review.

Include references to establish that you are familiar with prior work and how it relates to your work. Where possible/applicable, provide detailed performance data to establish that you have a working implementation and measurement tools. A good paper will demonstrate that the authors:

- are attacking a significant problem,
- are familiar with the current and past literature about the problem,
- have devised an original or clever solution,
- if appropriate, have implemented the solution and characterized its performance,
- have drawn appropriate conclusions about what they have learned.

Note that the USENIX Technical Conference, like most conferences and journals, requires that papers not be submitted simultaneously to more than one conference or publication, and that submitted papers not be previously or subsequently published elsewhere. Papers accompanied by non-disclosure agreement forms are not acceptable and will be returned to the author(s) unread. All submissions are held in the highest confidentiality prior to publication in the Proceedings, both as a matter of policy and in accord with the U.S. Copyright Act of 1976.

Authors will be notified by August 7, 1996. Some accepted papers will be shepherded by a program committee member through an editorial review process prior to publication in the conference proceedings.

Where to Send Submissions

Please send one copy of your manuscript to the program chairman via one of the following methods. All submissions will be acknowledged.

Preferred method:
email (PostScript or ASCII) to:
usenix97papers@usenix.org
If you have a MIME-capable mail system, you are encouraged to include your PostScript manuscript as Content-Type: application/postscript, Content-Transfer-Encoding: base64. Important: For PostScript submissions, use only standard PostScript fonts, and format your paper for US Letter (8.5 x 11 inches) paper. If your paper will not print properly, your submission will be returned. You should attach the cover letter (see below) as a separate MIME enclosure.

Alternate method:
Postal Delivery to:
John Kohl
Atria Software
20 Maguire Road
Lexington, MA USA 02173
Phone: 617.676.2641

The authors must also submit the following information (for administrative handling) via email to
usenix97papers@usenix.org
1. The title of the manuscript and the names of the authors. (Note: the program committee does not review papers blindly; the authors' names and affiliations will be known to the reviewers).
2. The name of one author who will serve as a contact, an email address, day and evening phone numbers, postal mail address, and a fax number, if available.
3. An indication of which, if any, of the authors are full-time students.
4. A short abstract of the paper (100-200 words) (this can be the same as the paper's abstract).

Cash Prizes

Cash prizes will be awarded for the best paper at the conference and the best paper by a full-time student.

Invited Talks

An Invited Talks track complements the Refereed Paper track. These talks by invited experts provide introductory and advanced information about a variety of interesting topics such as using standard UNIX tools, tackling system administration difficulties, or employing specialized applications. Submitted Notes from the Invited Talks are published and distributed free to conference technical sessions attendees. This track also includes panel presentations and selections from the best presentations offered at 1996 USENIX conferences and symposia.

Suggestions/Proposals Wanted

The Invited Talks coordinators welcome suggestions for topics and request proposals for particular talks. In your proposal, state the main focus, include a brief outline, and be sure to emphasize why your topic is of general interest to our community. Please submit via email to *ITusenix@usenix.org*.

Tutorials

On Monday and Tuesday, you may attend intensive, immediately practical tutorials on topics essential to the use, development, and administration of UNIX and UNIX-like operating systems, windowing systems, networks, advanced programming languages and related technologies. The USENIX Association's well-respected program offers introductory and advanced tutorials, presented by skilled instructors who are hands-on experts in their topic areas. USENIX will offer two full days of tutorials covering topics such as:

- System and network administration
- System and network security
- Java
- Distributed computing
- Kernel internals: SVR4, BSD, Windows NT
- Systems programming tools and program development

- Portability and interoperability
- Client-server application design and development
- Sendmail, DNS, and other networking issues
- GUI technologies and builders
- World-wide web technologies

Proposals Wanted

To provide the best possible tutorial slate, USENIX constantly solicits proposals for new tutorials. If you are interested in presenting a tutorial, contact the Tutorial Coordinator:

Daniel V. Klein
Phone: 412.421.2332
Email: *dvk@usenix.org*

Work-in-Progress Reports (WiPs)

Do you have interesting work you would like to share, or a cool idea that is not yet ready to be published? The Work-in-Progress reports, scheduled during the technical sessions, introduce interesting new or ongoing work. The USENIX audience provides valuable discussion and feedback. We are particularly interested in presentating student work. To schedule your report, send email to *wips97@usenix.org.*

Birds-of-a-Feather Sessions (BOFs)

The always popular evening Birds-of-a-Feather sessions are very informal attendee-organized gatherings of persons interested in a particular topic. BOFs often feature presentations or demonstration followed by discussion, announcements, and the sharing of strategies. BOFs are offered Tuesday, Wednesday, and Thursday evenings of the conference. BOFs may be scheduled on-site at the conference or in advance by contacting the USENIX Conference Office by phone at 714.588.8649 or via email to *conference@usenix.org.*

Vendor Exhibits

In the USENIX Vendor Exhibits, the emphasis is on serious questions and feedback. Vendors will demonstrate the technical innovations which distinguish their products. In this relaxed environment, conference attendees can discuss first-hand the product features and services on display. Plus, you can review the newest releases from technical publishers.

Vendors: This is an exceptional opportunity for receiving feedback from USENIX's technically astute conference attendees. If your company would like to display its products and services, please contact:

Zanna Knight
USENIX Association
Telephone: 510.528.8649
Fax 510.548.5738
Email: *display@usenix.org*

Conference Program and Registration Information

Special Hotel Rates

The Anaheim Marriott Hotel, adjacent to Disneyland, is headquarters for the USENIX 1997 Technical Conference and will be the location for all conference activities. The Anaheim Marriott will be offering special room rates to USENIX conference attendees.

Registration Materials

Materials containing all details of the technical sessions, tutorial program, conference registration, hotel and airfare discounts, and reservation information will be available mid-September, 1996.

If you wish to receive the registration materials, please contact:

USENIX Conference Office
22672 Lambert St., Suite 613
Lake Forest, CA USA 92630
Phone: 714.588.8649
Fax: 714.588.9706
Email: *conference@usenix.org*

About The USENIX Association

Since 1975, the USENIX Association has provided a forum where the community of engineers, scientists, and technicians working on the cutting edge of the computing world come together to communicate the results of innovation and research in UNIX and modern open systems. USENIX is well known for its technical conferences, tutorial programs, and the wide variety of publications it has sponsored over the years. USENIX is the original, not-for-profit membership organization for individuals and institutions interested in UNIX and related technologies. Evolving with technology, USENIX has broadened its activities to include open systems and the globally interconnected and interoperable computing environment.

The USENIX Association and its members are dedicated to:

- problem-solving with a practical bias,
- fostering innovation and research that works,
- rapidly communicating the results of both research and innovation, and
- providing a neutral forum for the exercise of critical thought and the airing of technical issues.

SAGE, the System Administrators Guild, a Special Technical Group within the USENIX Association, is dedicated to the recognition and advancement of system administration as a profession.

For more information about USENIX membership, SAGE membership, publications, or events, please visit the USENIX home page on the World Wide Web. Our URL is *http://www.usenix.org.*

Or, send email to our mailserver at *info@usenix.org.* Your message should contain the line: send catalog. A catalog will be returned to you.

How To Spot A System Administrator Who Hasn't Read Our New Books

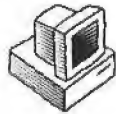**Weeps uncontrollably when asked "Want to go to lunch?"**

Thinks aspirin is a food group. **Speaks jibberish like "unix,**

shmunix." **Has a labrador retriever that goes by the**

name "Watchdog reset" **and howls ie bark**

Feeling a little frenzied? Gee, there are only a zillion things on your plate at any given time. But take heart: Where there's clear, hands-on information that helps solve your thorniest problems, there's relief. That's where our new system administration books come in.

Check out the new, second edition of our classic *Essential System Administration*, updated to include all the latest versions of major UNIX platforms. *Networking Personal Computers with TCP/IP* gives you information to tackle this sometimes daunting task.

And *Using and Managing UUCP* describes, in one volume, this popular communications and file transfer program. When it comes to security, we're publishing two books that will be essential: *Building Internet Firewalls* and *Computer Crime*. Both books are hands-on, current, and practical.

So, remember to take deep breathes and read the latest books from O'Reilly. To learn more about any of the books you see here, browse our online catalog at **http://www.ora.com/** And take a little time off when you can.
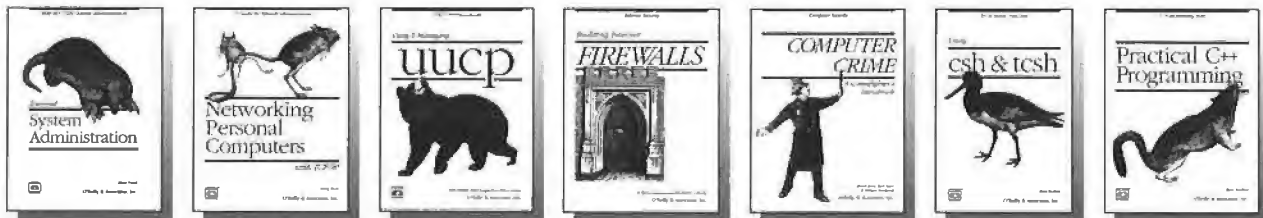
Announcing the *1996 Open Systems Products Directory*

The value of UniForum general membership is about to be enhanced with the publication of the *1996 Open Systems Products Directory.* You'll receive your personal copy with your paid membership (or renewal) to UniForum. Priced to sell to non-members at $150 per copy, the *1996 Open Systems Products Directory* is a real value. But, why spend an extra penny when it's **free** with paid membership?

Your $125 UniForum membership dues entitle you to a wealth of benefits and services, including: *IT Solutions* Magazine, *UniNews* Newsletter, UniForum Technical Publications, Member Discounts and the *1996 Open Systems Products Directory.* This entire package is available for one low fee. Call toll free (800) 255-5620 or (408) 986-8840 outside the U.S. Have your major credit card ready and we'll start your membership at once.
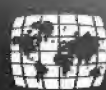
Now On-line!
Access the entire *1996 Open Systems Products Directory* over the World-Wide Web. UniForum members enter: http://www.uniforum.org

The *Open Systems Products Directory* has long been recognized as the most comprehensive guide to open systems products and services. It carries more than <u>8,700 detailed product and services listings</u> — more than ever before. With over 2,200 vendors, the Directory offers more than 1,900 pages of: Horizontal and Vertical Software, Network and Communications Software, Application Development Tools, System Hardware and Peripheral Devices, Operating Systems and Systems Software, and services such as Publications, Consultants, Conferences, Training and much, much more.

A collection of four detailed indexes: Vendors, Products, Keywords, and Operating Systems make the Directory indispensable. No wonder UniForum members renew year after year just to receive their latest edition. Users tell us they take it with them wherever they go on the job. The *1996 Open Systems Products Directory* quite simply is a resource tool you cannot afford to be without. It is yours free with UniForum membership.

2901 Tasman Drive, Suite 205
Santa Clara, CA 95054
(800) 255-5620
(408) 986-8840
Fax: (408) 986-1645
E-mail: pubs@uniforum.org
URL http://www.uniforum.org

UniForum
The International Association of Open Systems Professionals

ACM:
Association for Computing Machinery

ASPLOS:
Architectural Support for Programming Languages and Operating Systems

AUUG:
Australian UNIX Systems Users Group

COOTS:
Conference on Object-Oriented Technologies and Systems

DECUS:
Digital Equipment Computer Users Society

EurOpen:
European Forum for Open Systems

GURU: Romanian UNIX User Group

GUUG: German UNIX Users Group

HotOS:
Hot Topics in Operating Systems

IEEE: Institute of Electrical and Electronics Engineers

IETF:
Internet Engineering Task Force

INET:
Annual Conference of Internet Society

IWOOS:
International Workshop on Object-orientation in Operating Systems

JUS: Japan UNIX Society

LISA:
USENIX/SAGE Systems Administration Conference

OOPSLA:
Object-oriented Programming Systems, Languages and Applications

OSDI:
Symposium on Operating Systems Design & Implementation

POPL:
Principles of Programming Languages

ROSE: Open Systems in Romania

SANS:
System Administration, Networking & Security

SIGPLAN:
ACM Special Interest Group on Programming Languages

SIGSOFT:
ACM Special Interest Group on Software Engineering

SOSP:
ACM Symposium on Operating Systems Principles

SUG: Sun User Group

SUUG:
Society of Russia UNIX Users Group

UKUUG:
United Kingdom UNIX Systems Users Group

UniForum:
International Association of UNIX and Open Systems Professionals

WITI: International Network of Women in Technology

CALENDAR OF EVENTS

This is a combined calendar of conferences, symposia, and standards meetings. If you have an event that you wish to publicize, please contact *<login@usenix.org>*. For complete USENIX conference and symposia listings see URL *<http://www.usenix.org/events/general.html>*.

* = events sponsored by the USENIX Association.

1996

February
2 - 5   Freely Redistributable Software Conference, FSF, Cambridge, MA
14 - 16   UniForum, San Francisco, CA

March
4 - 8   IETF, Los Angeles, CA
27-30   Computers, Freedom & Privacy, Cambridge, MA

April
3 - 4   NetWorld+Interop '96, Las Vegas, NV
14-19   IEEE POSIX, Jackson Hole, WY

May
5 - 10   IEEE Symposium on Research and Privacy, Oakland, CA
13 - 17   SANS V, Washington, DC
18 - 24   DECUS, Orlando, FL
21-24   SIGPLAN '96, Philadelphia, PA

June
1 - 6   DECUS, '96 St. Louis, MO
10 - 14   NetWorld+Interop '96, Frankfurt, Germany
17 - 21 *COOTS II, Toronto, Canada
25 - 28   INET '96, Montreal, Canada

July
10 - 13 *Tcl/Tk, Monterey, CA
14 -19   IEEE POSIX, Nashua, NH
22 -25 * 6th UNIX Security, San Jose, CA
22 -26   NetWorld+Interop '96, Tokyo, Japan

August
4 - 9   SIGGRAPH, New Orleans, LA
5 - 9   Interex '96, San Diego, CA

September
3 - 5   GUUG, Leipzig, Germany
16 - 20   NetWorld+Interop '96, Atlanta, GA
30 -
Oct 4 *   LISA '96, Chicago, IL
         AUUG, Melbourne, Australia

October
1 - 4   ASPLOS VII, Cambridge, MA
6 - 11   OOPSLA '96, San Jose, CA
7 - 11   NetWorld+Interop '96, Paris, France

October (cont.)
8 - 10   UNIX Expo, New York City
23-25   IEEE Symposium on Reliable Distributed Systems, Niagara, Canada
27-28   IWOOOS '96, Seattle, WA
28 -31 * OSDI II, Seattle, WA
28 -Nov 1 NetWorld+Interop '96, London, England

November
4 - 8   Open Systems World/ FedUNIX Washington, DC
4 - 8   UNIX Network Security, Washington, DC
9 - 14   DECUS, Anaheim, CA
18 -20 * Electronic Commerce, Berkeley, CA
18 -22   ACM IEEE-CS Supercomputing '96, Pittsburgh, PA
25 - 29   NetWorld+Interop '96, Sydney, Australia

December
JUS UNIX Fair

1997

January
6 - 10 *   USENIX, Anaheim, CA
8 - 11 *   Linux, Anaheim, CA
20-24   ' POPL '97

March
12 - 14   UniForum, San Francisco, CA

April
7 -11   IETF, Memphis, TN

May
5 - 7   HotOS-VI

June
16 - 20   SIGPLAN '97

October
12 - 17   OOPSLA '97
27 - 31* LISA '97, San Diego, CA

1998

June
15 - 19 *   USENIX, New Orleans, LA
15 - 19   OOPSLA '98

December
7 - 11 *   LISA '98, Boston, MA
         JUS UNIX Fair

USENIX

1996 Conferences, Symposia, and Workshops for UNIX and Advanced Computing Systems Professionals

If these topics are important to you:

- Tcl/Tk
- UNIX Security and Applications of Cryptography
- Object-Oriented Technology
- Systems Administration
- Operating Systems Design and Implementation
- Electronic Commerce

then save these dates!

Plan to attend these USENIX events:

- **2nd Conference on Object-Oriented Technology.** June 17-21, 1996. Toronto
- **4th Annual Tcl/Tk Workshop.** July 10-13, 1996. Monterey, CA
- **6th UNIX Security Symposium Focusing on Applications of Cryptography.** July 22-25, 1996. San Jose, CA
- **10th Systems Administration Conference (LISA).** September 30-October 4, 1996. Chicago
- **2nd Symposium on Operating Systems Design and Implementation (OSDI).** October 28-31, 1996. Seattle, WA
- **2nd Electronic Commerce Workshop.** Dates and location to be announced.

For more information:

USENIX, 22672 Lambert St., Suite 613,
Lake Forest, CA 92630
Phone: 714.588.8649
Fax: 714.588.9706
Email: *conference@usenix.org*
WWW: *http://www.usenix.org*

USENIX, the UNIX and Advanced Computing Systems Technical and Professional Association, offers technical conferences for and by technical professionals.
SAGE, the System Administrators Guild, a special technical group within USENIX, is dedicated to the advancement and recognition of system administration as a profession.

;login:

**USENIX Association
2560 Ninth Street
Suite 215
Berkeley, CA 94710**

POSTMASTER:
SEND ADDRESS CHANGES
TO *;login:*
USENIX ASSOCIATION
2560 NINTH STREET
SUITE 215
BERKELEY, CA 94710

D